



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**SISTEMA DE TRANSMISIÓN DE ARCHIVOS VIA
NDM SOBRE PLATAFORMA MAINFRAME**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A:
RODRIGO RODRÍGUEZ ESTUDILLO

ASESORA: ING. SILVIA VEGA MUYTOY



MÉXICO

2006



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



Mi sincero agradecimiento:

A Dios por permitirme lograr mis metas.

A mis padres por su apoyo, por su ejemplo y por los valores que me inculcaron.

A mis hermanos por sus valiosas enseñanzas, consejos y apoyo.

A Laura Liceaga por su amor y cariño.

A mi Alma Mater, la UNAM por darme la oportunidad de ser competitivo en el mercado laboral.

A Silvia Vega por ser además de una gran maestra, una excelente persona.

A Jesse Méndez por su calidad humana y su amistad.

A todos mis profesores por la formación académica que me han dado.



ÍNDICE

	Página
OBJETIVO.....	viii
INTRODUCCIÓN.....	x
CAPÍTULO 1	
1 ANTECEDENTES.	
1.1 La plataforma Mainframe.....	1
1.2 El sistema operativo MVS.....	6
1.3 El ambiente TSO (Batch).....	9
1.4 El monitor de transacciones CICS (línea).....	13
1.5 El tipo de dato Queue en CICS.....	16
1.6 Bases de datos con DB2.....	18
1.7 El lenguaje de programación COBOL CICS.....	30
1.8 La tecnología de transmisión NDM.....	38
1.9 El modelo de redes OSI y el protocolo TCP/IP.....	42
1.10 El modelo de negocio para transmisión de archivos.....	45
CAPÍTULO 2	
2 ANÁLISIS DEL PROBLEMA.	
2.1 Delimitación del problema.....	47
2.2 Factibilidad.....	51
2.3 Alcances del sistema.....	52



INDICE

	Página
CAPÍTULO 3.	
3 DISEÑO Y DESARROLLO DEL SISTEMA.	
3.1 Diagrama conceptual del sistema.....	56
3.2 Definiendo y estructurando el JCL de transmisión.....	57
3.3 Definiendo los parámetros que el sistema recibirá.....	58
3.4 Diseño de la pantalla (Front de transmisión).....	59
3.5 Diagrama de estados de la aplicación en línea.....	64
3.6 Seguridad del front de transmisión.....	66
3.7 Diagrama de flujo de la aplicación Batch.....	67
3.8 Diseño de la entidad que almacena acciones y rutas.....	68
CAPÍTULO 4.	
4 PRUEBAS, LIBERACIÓN Y MANTENIMIENTO.	
4.1 Prueba unitaria.....	73
4.2 Prueba modular.....	76
4.3 Consideraciones para aumentar la funcionalidad.....	78
4.4 Consideraciones para migrar el sistema a otros equipos.....	79
CONCLUSIÓN GENERAL.....	81
GLOSARIO.....	82
BIBLIOGRAFÍA.....	84
RESUMEN, TABLAS Y FIGURAS.....	85



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

OBJETIVO GENERAL



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

OBJETIVO

El presente trabajo propone la solución para las instituciones financieras que buscan y necesitan distribuir su información de operación entre sí mismas y con los organismos gubernamentales que rigen sus procedimientos y funciones.

Generalmente el usuario depende del personal de sistemas para realizar las tareas de envío de archivos a las instituciones de regulación bancaria, esta dependencia se forma debido a que los procesos que se utilizan para este propósito son, por lo general, extremadamente complejos y restringidos para él.

La dependencia descrita no sólo causa multas gubernamentales hacia las instituciones bancarias por entregas fuera de horario (debido a que la persona de sistemas no atendió al usuario en el horario establecido), también causa ineficiencia y deteriora la calidad de la operación.

En este proyecto, se describe e ilustra la manera en que se resuelve el problema en cuestión, desarrollando un sistema amigable que sea capaz de ejecutar todas las tareas necesarias para la transmisión de archivos. El usuario podrá realizar la transmisión de archivos con oportunidad y calidad sin depender del departamento de sistemas.

JUSTIFICACIÓN

Existen varios problemas por resolver en la operación de una institución financiera, el primero y más importante es la mala distribución de la información, tómese el ejemplo de una AFORE¹, si esta no informa en tiempo, forma y contenido a la CONSAR² sobre sus operaciones, se hará acreedora a una fuerte multa. Esto provoca diversas consecuencias, entre ellas la mala imagen como administradora de fondos para el retiro o en muchos casos, afecta directamente la respuesta a alguna petición hecha por sus participantes o afiliados. Este problema ocurre principalmente debido a que el usuario no cuenta con una herramienta amigable que le permita enviar y recibir archivos hacia quien lo requiera, y depende directamente de un sistema automático que se ejecuta por la noche o bien, del personal de sistemas para un envío o recepción manual.

¹ Administradora de Fondos para el Retiro. Las Afores son empresas financieras debidamente autorizadas por la SHCP, y supervisadas por la CONSAR, que se especializan en el manejo de los ahorros para el retiro de los trabajadores.

² Comisión Nacional del Sistema de Ahorro para el Retiro.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

BREVE DESCRIPCIÓN

Este proyecto fue planteado a partir de los objetivos y requerimientos de un organismo financiero para agilizar y optimizar sus operaciones.

En el capítulo I se presenta un panorama general de la plataforma y la tecnología que generalmente Bancos, Afores, Casas de bolsa y Aseguradoras utilizan para la tarea de enviar y recibir archivos. La plataforma se refiere a IBM³ OS/390⁴ Mainframe, la Tecnología se basa en el manejador de bases de datos de IBM DB2⁵, al lenguaje de programación COBOL⁶, NDM para transmisión de archivos y el manejo de transacciones al CICS⁷.

En el capítulo 2 se cubren temas relacionados con el análisis del problema, tales como la identificación de las necesidades de una AFORE en cuanto al envío y recepción de información, los objetivos específicos, el plan de actividades y la viabilidad del proyecto.

El capítulo 3 se ocupa de mostrar el diseño del sistema, como los programas, los procesos, la interfaz con el usuario y por supuesto, la base de datos. El diseño es plasmado claramente en diagramas de estado y de flujo en el caso de programas y procesos, en el caso de tablas de base de datos es presentado en diagramas entidad-relación.

En el capítulo 4 se detalla la forma en que se realizan las pruebas de un sistema como este hasta llegar a la liberación y el mantenimiento del mismo. También se puede encontrar un breve apartado que define la posibilidad de realizar un sistema idéntico, en otra plataforma.

³ International Business Machines. Empresa precursora de la industria de la computación.

⁴ Sistema Operativo de la familia de supercomputadoras Mainframe de IBM.

⁵ Manejador de Bases de Datos de la firma IBM.

⁶ Acrónimo de Common Business Oriented Lenguaje, es un lenguaje de programación orientado a las finanzas y es popular por su poder para explotar las bases de datos.

⁷ Acrónimo de Customer Information Control System. Es un monitor de transacciones capaz de atender miles de peticiones por segundo.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

HIPÓTESIS

La problemática que presentan las instituciones financieras, en concreto una AFORE, al transmitir información, sirve para hacer el análisis de un sistema de tal forma que se convierta en la solución que ayude a mejorar la operación al enviar y recibir archivos de los organismos que validan sus procesos internos. La AFORE requiere de un sistema que le permita transmitir archivos en cualquier horario (en línea), con una interfaz amigable y de gran confiabilidad, que permita la independencia del usuario con el personal de sistemas. Esto permitiría mejorar la operación y evitar multas por entregar información fuera de tiempo.

Para llegar a esta solución técnica, es necesario emplear las metodologías mencionadas por las tecnologías de información que cubran las etapas por las que pasa todo sistema informático: Análisis, Diseño, Desarrollo, Pruebas, Implementación y Mantenimiento. A lo largo del presente proyecto se apreciara cada una de estas etapas para el desarrollo y liberación del Sistema de Transmisión de Archivos vía NDM sobre plataforma Mainframe (STA).



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

INTRODUCCIÓN



INTRODUCCIÓN

La información es de esencial importancia a lo largo de la vida de los individuos, es ella quien proporciona los elementos para la toma de decisiones a largo y a corto plazo, influyendo significativamente en las acciones de cada persona.

La necesidad de que exista una técnica que se encargue de resguardar y administrar la información se torna indispensable. La informática es la técnica y disciplina con la que se pueden realizar estas tareas, pero ¿de qué sirve contar con la información si no se tiene una herramienta con la que podemos distribuirla de manera oportuna a las personas indicadas?

Las dimensiones de la cuestión anterior son de carácter exponencial cuando en lugar de hablar de individuos, hablamos de una organización del tipo de las instituciones financieras.

Las actividades de cualquier organización están influenciadas por la capacidad de ésta para transmitir datos con rapidez y eficiencia. El sistema de información que este trabajo presenta: Sistema de transmisión de archivos vía NDM¹ sobre plataforma Mainframe² (denominado en lo sucesivo STA³) es un sistema de información que mejora esta capacidad en tres formas.

Aumenta la velocidad de procesamiento. Los sistemas basados en computadora pueden ser de ayuda para eliminar la necesidad de cálculos tediosos y comparaciones repetitivas. Un sistema automatizado puede ser de gran utilidad si lo que se necesita es un procesamiento acelerado. Esta Tesis propone que se desarrolle una aplicación para que el usuario pueda por sí mismo transmitir archivos a la institución gubernamental que se lo requiera, sin depender del personal de sistemas en cada envío y/o recepción.

Aumento en el volumen. La incapacidad para mantener el ritmo de procesamiento no significa el abandono de los procedimientos existentes. Quizá éstos resulten inadecuados para satisfacer las demandas actuales. En estas situaciones se considera el impacto que tiene la introducción de procesamiento computarizado, ya que el sistema existente es manual. Es poco probable que únicamente el aumento de la velocidad sea la respuesta. El tiempo de transmisión de datos aumenta si se considera la cantidad de actividades financieras de la empresa junto con su patrón de crecimiento.

Transmisión más rápida de la información. Las organizaciones almacenan grandes cantidades de datos, por eso, debe tenerse en cuenta dónde almacenarlos, cómo recuperarlos y distribuirlos cuando se les necesita. Cuando un sistema se desarrolla en forma apropiada, se puede recuperar en forma rápida la información, objeto de estudio de este texto es, transmitir eficientemente dicha información. Como parte de la solución se emplea la tecnología basada en TCP/IP y la herramienta NDM para realizar la transmisión de datos.

¹ Network DataMover. Producto para transmisión de datos basado en el protocolo de TCP/IP.

² Arquitectura de supercomputadoras introducida por IBM Corp.

³ Sistema de transmisión de archivos.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

Este trabajo trasciende a la habilidad para diseñar, desarrollar y optimizar un sistema capaz de transmitir archivos entre los organismos financieros, basándose en la arquitectura de la plataforma Mainframe y el sistema operativo OS/390.

En la construcción y desarrollo de este proyecto se aplican métodos y técnicas para resolver los problemas que tienen las instituciones financieras para transmitir información, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

CAPÍTULO 1



CAPÍTULO 1

El entorno que corre actualmente en los modernos centros de proceso de datos es muy parecido al de hace 20 años, salvo que ahora todo es más grande, más complejo y mucho más rápido.

En la historia de los sistemas operativos que han corrido sobre el sistema 360 o el 370 de IBM se ha respetado el hardware, porque todavía siguen funcionando muchos de los equipos antiguos.

Si se estudian sus orígenes, se comprenderá más fácilmente lo que este trabajo de tesis pretende mostrar, trazando un rumbo para llegar a nuestro destino: Crear una aplicación de 370 para enviar y recibir archivos.

1.1 La plataforma Mainframe.

La arquitectura de sistemas es el término utilizado para describir la relación entre las partes de la computadora y el sistema operativo que se ejecuta en la maquina.

En el verano de 1964 la arquitectura del sistema 370 vio la luz. Todos los modelos del 370 estaban orientados a números binarios, decimales, en punto flotante de doble precisión, a caracteres y palabras, todo en una única y rentable computadora. El sistema 370 prometía desde su introducción en el mercado compatibilidad con las anteriores y siguientes versiones de la plataforma Mainframe. La compatibilidad es prácticamente absoluta, sin embargo, existen limitaciones:

- El programa es independiente de las facilidades operativas que no estén incluidas en el siguiente modelo de hardware más veloz.
- El programa es independiente de instrucciones ejecutadas en un periodo de tiempo concreto.
- El programa es independiente del hardware que no este incluido en la nueva versión del sistema.
- El programa es independiente de las facilidades incluidas en el nuevo sistema después de la creación del programa (el programa no admitirá las interrupciones causadas por instrucciones con códigos de operación inválidos, si estos están presentes en la nueva versión del sistema).
- El programa es independiente de los resultados o funciones que están definidas en el manual de principios de operación del nuevo modelo, las cuales son impredecibles de antemano.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

Como principales características del sistema 370 podemos citar las siguientes:

- Traducción dinámica de direcciones (DAT). Con esta característica el programador de aplicaciones se libera de la preocupación por todo lo concerniente al hardware en donde se esta ejecutando la aplicación.
- Multiprocesamiento. El MVS¹ esta diseñado para sistemas multiprocesadores. Las aplicaciones pueden sacar provecho de varias CPU's² en una misma computadora que, bajo MVS, trabajan al máximo rendimiento.
- Facilidades de protección. Las aplicaciones pueden beneficiarse de las mejoras en las facilidades de protección que incorpora el sistema 370.
- Direccionamiento real extendido. Las computadoras sistemas 370 están diseñadas para eliminar las consideraciones sobre la memoria central de la aplicación.
- Canal de direccionamiento indirecto. El sistema 370 tiene muchas posibilidades diferentes para canales, y el canal de direccionamiento indirecto le permiten desarrollarse durante los años venideros.

Sistemas operativos.

La meta de un sistema 370 era la de capacitar el uso de un mismo sistema operativo en configuraciones de hardware grandes y pequeñas. Los creadores de software crearon, por otra parte, varios sistemas operativos para el mismo 370.

Los sistemas 370 han sobrevivido a la promesa. Generalmente, el software ha seguido siendo compatible para las máquinas que han ido apareciendo, pero, como se verá más adelante, la conversión desde el DOS³, pasando por el MVT⁴, hasta el MVS era complicada. Los problemas no surgían con los programas de aplicación, sino en las áreas de sistemas y operaciones.

Los primeros sistemas operativos del 370.

Para situar el MVS en su contexto histórico, se mencionan algunas alternativas a este y a sus predecesores. Estos sistemas operativos estaban diseñados para ser ejecutados en computadoras mucho más pequeñas que aquellas para las que estaba concebido el MVS, pero su evolución es la base para muchas de las funciones que hoy permite éste.

¹ Memoria Virtual Múltiple, el MVS se torna el sistema operativo de mayores prestaciones para el Mainframe.

² Unidad central de proceso de una computadora.

³ Disk Operating System, el primer sistema operativo en disco.

⁴ Multiprogramación con número variable de tareas.



Sistema operativo en cinta.

A principios de los sesenta, los denominados equipos Mainframe, utilizaban sistema operativo en cinta (TOS). El sistema operativo se cargaba en la computadora desde una cinta y el centro de cálculo podía entonces ejecutar todo aquello que había desarrollado para procesar sus tareas. Casi todos los programas eran escritos a medida por el centro de cálculo para sus necesidades específicas.

Sistema operativo en disco.

A principios de los setenta, el Disk Operating System (DOS) fue utilizado para ejecutarse en las pequeñas computadoras de la gama de los 370. El disco que contenía el software de sistema operativo era un dispositivo de memoria de acceso directo (DASD). Por medio del DOS el centro de cálculo podría tener dos aplicaciones ejecutándose al mismo tiempo (Foreground y Background). Se utilizan dos particiones porque una vez que se empieza a ejecutar una sola tarea en una partición, se tienen recursos (Operadores, DASD, unidades de disco) parados y sin producir. En muchos lugares el Foreground ejecutaba una aplicación de Terminal interactiva y en Background se ejecutaban Test y tareas de producción.

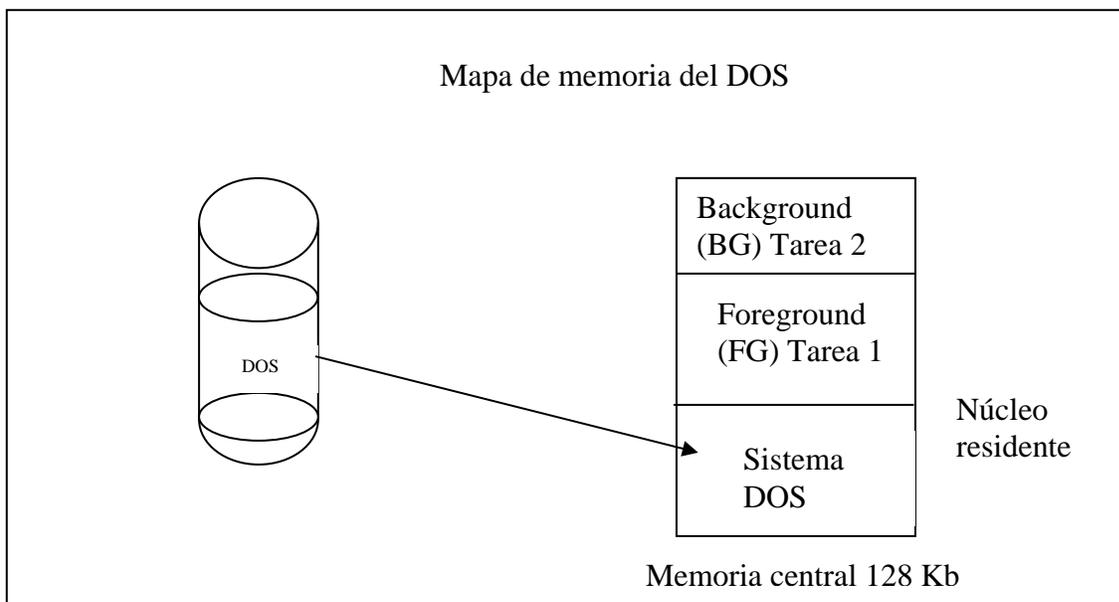


Figura 1.1: Mapa de Memoria del DOS



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

La figura anterior muestra como el DOS tiene un diminuto sistema operativo y dos particiones. Cada partición puede soportar la ejecución de una tarea. El DOS es un núcleo residente que contiene los programas del sistema operativo. También cuenta con área para que DOS pueda cargar, de forma dinámica, módulos que contienen funciones adicionales.

Tipos de sistemas operativos.

A la vez que las computadoras aumentaban de tamaño, el Hardware empezó a ser capaz de ejecutar más de dos aplicaciones a la vez. Varias versiones de sistemas operativos fueron concebidas para ser utilizadas en las máquinas grandes.

Control primario de programas (PCP). El primer sistema operativo después del DOS, con este sistema operativo se podían ejecutar aplicaciones de una en una, se arrancaba un programa, cuando terminaba se podía empezar otro. El PCP introduce los servicios de supervisor que todavía pueden encontrarse en el MVS actual.

Con el PCP ningún programa es jamás retrasado por otro, fue uno de los últimos sistemas operativos para Mainframe capaz de hacerlo.

Los términos multitarea o multiprogramación son utilizados para referirse a cuando se tiene la posibilidad de concederles a varios programas de aplicación los recursos del sistema.

Los sistemas multitarea retrasarían una simple aplicación. Se puede dar prioridad a ciertas aplicaciones, pero puesto que el sistema operativo realiza otras tareas, incluso la máxima prioridad será ligeramente demorada.

Multiprogramación con número fijo de tareas (MFT). La mayor ventaja del MFT es que el centro de cálculo puede ejecutar más de una aplicación al mismo tiempo.

La parte fija de este sistema operativo se debe a que en la instalación se definen particiones de tamaño fijo para llevar a cabo el trabajo. La primera partición es llamada P0 y la segunda P1, y así sucesivamente hasta un máximo de 15. En la arquitectura del 370 siempre el primero de la lista será el elemento cero.

Multiprogramación con número variable de tareas (MVT). Un tercer tipo de sistema operativo OS es el MVT. Bajo MVT el centro de cálculo no tiene que decidir de antemano el tamaño de los programas que podían ser ejecutados. El programador especifica el tamaño del programa, y el MVT reserva la región de ese tamaño.

Es muy poca la diferencia entre una región y una partición. El MVT es el sistema más popular en los equipos 360 y 370, hasta que su evolución lo llevo a convertirse en el MVS. La figura 1.2 muestra el aspecto de los equipos 360 y 370, mientras que, la figura 1.3 muestra la organización y funcionamiento del sistema 360.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"



Figura 1.2: Equipo Mainframe de IBM corriendo n sistema operativo 360.

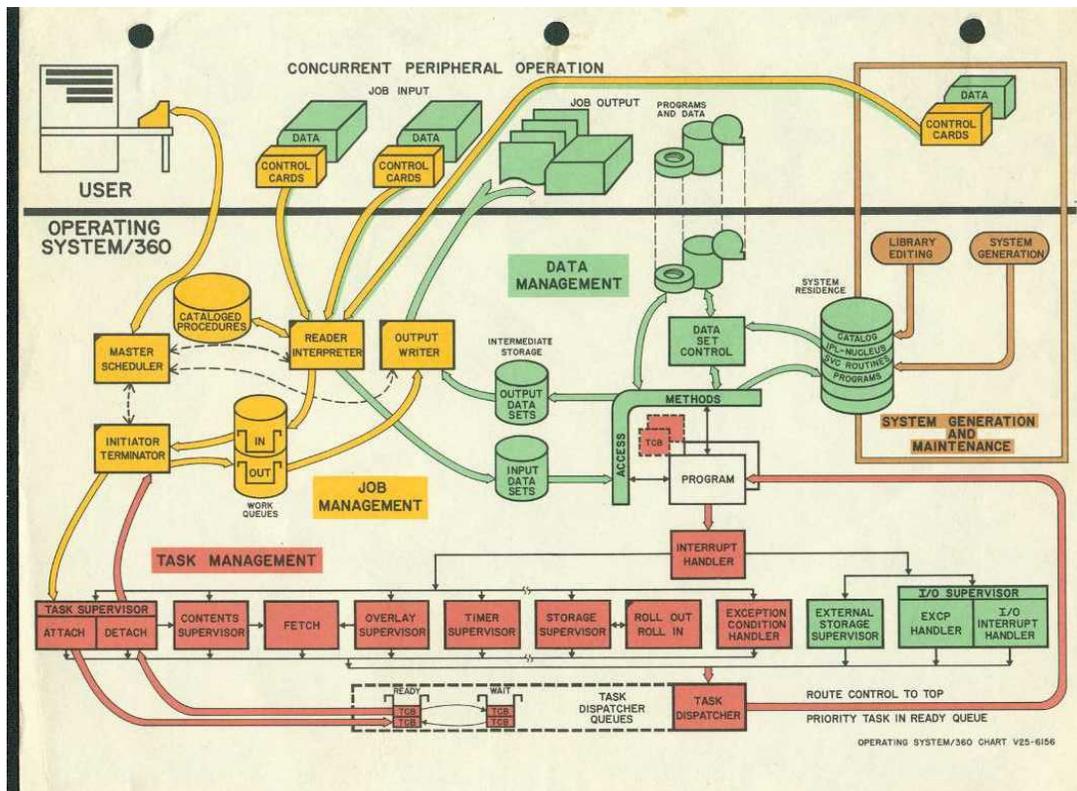


Figura 1.3: Diagrama a bloques que representa la organización del sistema operativo 360 .



1.2 El sistema operativo MVS.

Los ingenieros de hardware habían conseguido construir computadoras que eran más grandes y más rápidas de que los creadores de software podían utilizar con efectividad.

La primera restricción era el insuficiente espacio de memoria central que se podía asignar a algunas grandes aplicaciones, la razón de esto es que dos áreas de memoria no direccionadas no son contiguas. Esto denomina **fragmentación de memoria**. La fragmentación de memoria es de los primeros problemas resueltos por los sistemas operativos virtuales.

Existe otro problema que se resuelve con MVS, si el programa a ejecutarse necesitaba 1Mb de memoria y la computadora sólo tenía medio Mb instalada, entonces no se podía ejecutar el programa. El aspecto que resuelve este problema es precisamente la memoria virtual. Se crearon diversos sistemas operativos virtuales:

DOS/VS y VSE. Los creadores del DOS añadieron la memoria virtual a sus sistemas. DOS/VS fue la primera versión. La versión mejorada fue el DOS/VSE y hoy todavía sigue siendo utilizado.

VS1: Almacenamiento virtual 1. Sistema operativo similar al MFT anterior, sólo que con memoria virtual. El operador puede redefinir las particiones dinámicamente. Se pueden utilizar hasta quince particiones de programas para ejecutar trabajos. El VS1 soporta el CICS⁵ y el IMS⁶ para funciones del proceso de transacciones interactivas u Online. VS1 tiene un alto grado de compatibilidad con el MVS y proporciona un atractivo entorno para procesos.

VS2: Memoria virtual simple (SVS). La memoria virtual 2 es muy similar al MVT, pero con memoria virtual. El sistema operativo gestionaba dinámicamente la memoria virtual, el operador no tenía que establecer particiones.

La fragmentación de memoria deja de ser también un problema porque la aplicación esta hecha para considerar que la memoria es contigua, aunque esta esparcida a lo largo de la memoria central. Es el sistema operativo, y no el programa de aplicación, quien puede gestionar los recursos.

El sistema operativo logra esto partiendo del fundamento de que ningún programa necesita la memoria durante todo el tiempo, por lo que el sistema operativo divide la memoria en bloques del mismo tamaño llamados páginas, entonces se asegura de que sólo las páginas requeridas entren a la memoria central, mientras el resto se graba en una DASD. Intercambiando páginas entre la memoria central y el DASD en un complicado juego, el sistema operativo hace que parezca que hay más memoria central de la que realmente existe. El sistema operativo puede ahora ejecutar más programas que los propios de la memoria central.

⁵ Customer Information Control System, monitor de transacciones para administrar concurrencias en el sistema.

⁶ Information Manager System, dentro del sistema operativo pieza fundamental para la óptima administración de la información.



VS2: Memoria Virtual Múltiple (MVS). El MVS introduce el concepto de **espacio de direcciones**. El espacio de direcciones es el rango completo de las direcciones de la computadora que un programa puede utilizar. El programa puede ser un proceso batch, una tarea de terminal o una tarea del sistema. El término de identificador de espacio de dirección (ASID) se utiliza para identificar un espacio de dirección.

Cada espacio de direcciones o programa de aplicación puede acceder a un máximo de 16 Mb de memoria, sin contar en absoluto con el tamaño de la memoria central. Esto es distinto del SVS, donde la suma de todos los espacios de direcciones no puede exceder de 16 Mb. El límite es elevado por el MVS/XA a dos Gb y a 16 Tb por el MVS/ESA.

MVS: Emisiones 1 y 2. La versión 1 de MVS fue la primera de los sistemas operativos MVS. Tenía varios problemas de seguridad.

El sistema podía estar trabajando durante unas horas y después había que reiniciarlo, pero el MVS daba un buen número de prestaciones a los centros de cálculo:

1. Tareas ilimitadas.
2. Memoria virtual ilimitada. Los problemas de gestión de particiones y de cambio de un trabajo por otro habían terminado.
3. Menos ordenes de operador que envolvían la estructura del entorno del sistema operativo. El MVS trajo técnicas de gestión imprescindibles en un centro de cálculo grande.

MVS Versión 3. En 1979 fue anunciada la versión 3.8 de MVS. Esta versión también tenía sus problemas, puesto que los problemas son algo que se descubre y se soluciona cada día. La versión 3.8 de MVS se ve beneficiada por las mejoras que le proporcionan las unidades opcionales (SU's), que son modificaciones o reemplazos para su óptimo funcionamiento. Algunas de las más importantes fueron:

1. SU 04: Listado de mejoras, permitía más eficacia en el trabajo.
2. SU 05: Supervisor de rendimiento, mejoraba el rendimiento de las rutinas del supervisor de MVS.
3. SU 09: Método de acceso secuencial extendido, mejoraba el rendimiento de accesos secuencial a archivos. Principalmente incrementando el número por defecto de los buffer E/S de dos a cinco.
4. SU 60: Esta mejora colocaba la fecha en la entrada VTOC de cada conjunto de datos a los que accedía MVS. La fecha se utilizaba para guardar el último día que un conjunto de datos había sido utilizado por un programa. Los programas de gestión de DASD aún utilizan este sistema para archivar
5. conjuntos de datos a los que no se ha accedido durante un periodo de tiempo específico.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Extensiones del sistema MVS. El MVS se transformó en el MVS/SP que era el sistema operativo MVS con las mejoras obtenidas con el SP (grupo de mejora al sistema operativo System Product), esta versión también es conocida como MVS/370. Esta versión soporta la arquitectura extendida; en vez de 16 Mb de espacio de direcciones la de 2 Gb.

Como el proceso de datos necesita ser ampliado, la necesidad de acceder a mas datos requería procesos más complejos, más DASD online, esto llegó a ser más importante para las corporaciones que la facilidad de acceso a los datos.

MVS SP VERSIÓN 3-. Arquitectura de sistemas de empresa (Enterprise System Architecture) Cuenta con los mismos 2 Gb de espacios de direcciones disponibles para programas y datos que había en la versión anterior. La novedad es que hay hasta un máximo de 15 espacios de direcciones de 2Gb más, exclusivamente para datos.

Muchas de las mejoras vienen en el micro código, las instrucciones en la computadora que realizan el trabajo que las órdenes de la aplicación necesitan hacer.

Otros sistemas operativos. El MVS no es el único sistema operativo que soportan los sistemas 370. Se puede citar entre estos sistemas operativos a las máquinas virtuales.

La producción en tiempo compartido utiliza el concepto de máquina virtual (VM). El VM juega un importante papel en el desarrollo del MVS y debe ser mencionado con la misma profundidad que el propio MVS. El MVS puede ejecutarse en una máquina virtual bajo VM. El uso correcto de este entorno hace que el MVS sea muy productivo. Cada aplicación bajo el VM/370 tiene control de sus propios espacios de direcciones de 16 MB o 2Gb con el VM/XA. Cada uno de estos sistemas operativos son referenciados como sistemas operativos huéspedes.

Esto hace del VM una importante herramienta de pruebas y producción. Por ejemplo, una VM podría emular a un MVS/370 en un ambiente de pruebas. Es decir, este sistema operativo VM, podría incluso emularse a sí mismo.

El MVS es uno de los mejores sistemas operativos jamás creados. Una de las razones de esto es que se han satisfecho las necesidades del usuario final. En un principio el usuario final se conformaba con encontrar al día siguiente los reportes que necesitaba, posteriormente podía acceder a ellos en línea desde un terminal en su despacho, ahora además de ello, necesitan de una forma segura y eficiente de distribuir esa información.

El CICS y el IMS controlan las terminales y las bases de datos.

La figura 1.4 representa una clásica configuración del sistema MVS.

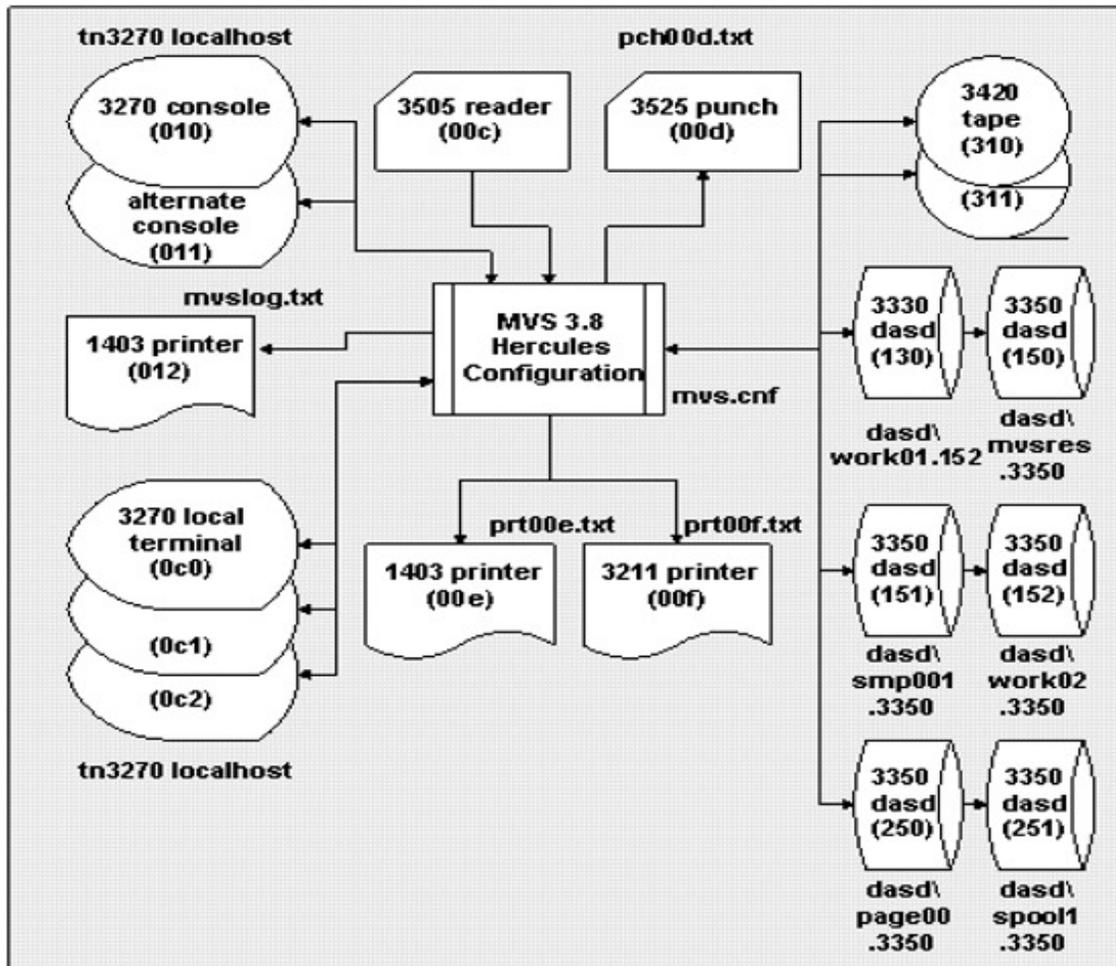


Figura 1.4: Configuración clásica de un sistema MVS.

1.3 El ambiente TSO (Aplicaciones Batch).

El familiarizarse con el ambiente TSO (Time Sharing Options) resulta muy importante debido a que es con esta herramienta como el operador interactúa con el MVS y sus aplicaciones. El TSO es un software que al ejecutarse sobre MVS proporciona el menú para que el programador u operador pueda realizar las distintas tareas que ejecuta.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

Esas tareas son generalmente: La creación y manipulación de programas con los comandos básicos, archivos de trabajo, procedimientos y JCL's⁷, monitoreo de tareas (jobs) en ejecución, entre otras.

Una Terminal de Mainframe puede acceder a las funciones del mismo a través de una sesión de TSO. La autenticación del usuario que se conecta mediante TSO corre a cargo del administrador del equipo quien asigna acceso y privilegios mediante la utilería conocida como RACF⁸.

Con el TSO se puede ejecutar comandos y acceder a utilerías a través de su menú, existen teclas de control que son utilizadas para obtener un mejor desempeño en las tareas de consulta o edición de componentes. Estas teclas de control tienen un valor estándar aunque también pueden ser personalizadas configurándolas.

Los comandos que están disponibles a través de TSO (Tso es el menú y la línea de comandos y tareas para explotar los recursos del sistema) son los siguientes:

ISPF primary Option Menu. Es el menú principal de TSO, desde aquí se puede acceder a todas las utilerías del sistema, como son:

- **View Display source data or listings:** Esta opción nos permite utilizar el editor del sistema desde donde podemos trabajar con archivos o con código de programas. El modo VIEW no es de edición. Es importante mencionar que existen dos tipos de archivos: MEMBER y DATA SET. Se denomina MEMBER (miembro) a cada uno de los componentes que se encuentran en una biblioteca o archivo particionado (programas, copys, jcls, etc.) y se conoce como DATA SET a los archivos de trabajo que contienen datos que serán manipulados por un proceso. Esta opción es pues útil, cuando se desea consultar un componente sin modificarlo.
- **Edit Create or change source Data:** En esta opción se puede crear y modificar el contenido de los componentes miembro y de los archivos de trabajo.
- **Utilities:** Dentro de este menú se encuentran las utilerías para el manejo de archivos de las cuales las más comunes son: comprimir librerías, renombrar, borrar, editar, mover, copiar e imprimir archivos.
- **Foreground:** Esta opción permite el procesamiento interactivo de un JCL.
- **Batch:** Desde aquí es posible submitir (lanzar) a ejecución un job.
- **Command:** Permite ejecutar comandos TSO.
- **DB2:** Es el menú correspondiente a la Base de Datos DB2.

El ambiente TSO y los programas Batch.

Existen dos técnicas para manipular la información: Batch y Online.

⁷ Job Control Lenguaje. Procedimientos conformados por pasos para ejecutar distintas tareas en un proceso automatizado.

⁸ Utilería presente en MVS y esta destinada a gestionar la seguridad.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

La técnica Batch modifica la información con cierta periodicidad, puede ser diaria, semanal, quincenal, mensual, etc., pero nunca lo hace justo en el momento en que los datos sufren modificaciones, de hecho, los procedimientos Batch carecen de interfaz de usuario. La técnica Online modifica la información en tiempo real, más o menos cuando los datos cambian y cuentan con interfaz de usuario ya que esta ligada estrechamente a él. En este apartado se estudiara la técnica Batch.

Como se puede observar, el TSO es el intérprete que permite enviar comandos al MVS para que este los ejecute, no solo comandos se pueden procesar, también programas escritos bajo la tendencia Batch.

Un programa batch es aquel que ha sido escrito para ejecutarse generalmente durante la noche, este código será capaz de procesar grandes cantidades de información y no necesita la intervención de un usuario para cumplir con su tarea. En otros términos, el programa Batch recibe datos, los procesa y genera una respuesta de salida sin interactuar con un usuario.

Los programas batch en plataforma Mainframe son, por lo general, código en COBOL y con sentencias incrustadas SQL para DB2. Estos programas son disparados o ejecutados a través de un JCL.

El JCL (Job Control Lenguaje) es un procedimiento constituido por pasos de ejecución, cada paso puede estar definido por una utilería propia del sistema (como una descarga de información o la creación de un archivo de trabajo) o por una tarea aplicativa (programa COBOL).

El programa Batch se ocupa para el procesamiento en lotes y es preciso que considere lo siguiente:

- Definir y signar valores a las áreas de datos y variables.
- Abrir todos los archivos de trabajo.
- Leer datos de entrada.
- Procesar datos leídos y actualizar el archivo Maestro.
- Imprimir el registro actualizado en el informe de inventario.
- Crear en caso de ser necesario, un archivo de salida.
- Cerrar todos los archivos de trabajo cuando el programa haya terminado de procesar.

En el procesamiento en lotes, el programa ha de atender con frecuencia diferentes tipos de transacciones de entrada. En ocasiones, el resultado es un programa largo y complicado. Además, como en un momento dado se procesa todo un lote de transacciones, los usuarios deben disponer de su información de entrada en un momento y lugar designados. Luego, el programa de aplicación pide que los datos sean leídos en memoria, y los datos son procesados. Después de procesados los datos, la salida impresa se devuelve a los departamentos de usuarios. El tiempo transcurrido entre la información de entrada y la



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

información de salida impresa pueden ser horas o días. Una preparación de reportes, es siempre un proceso Batch. La figura 1.5 muestra el aspecto del ambiente TSO.

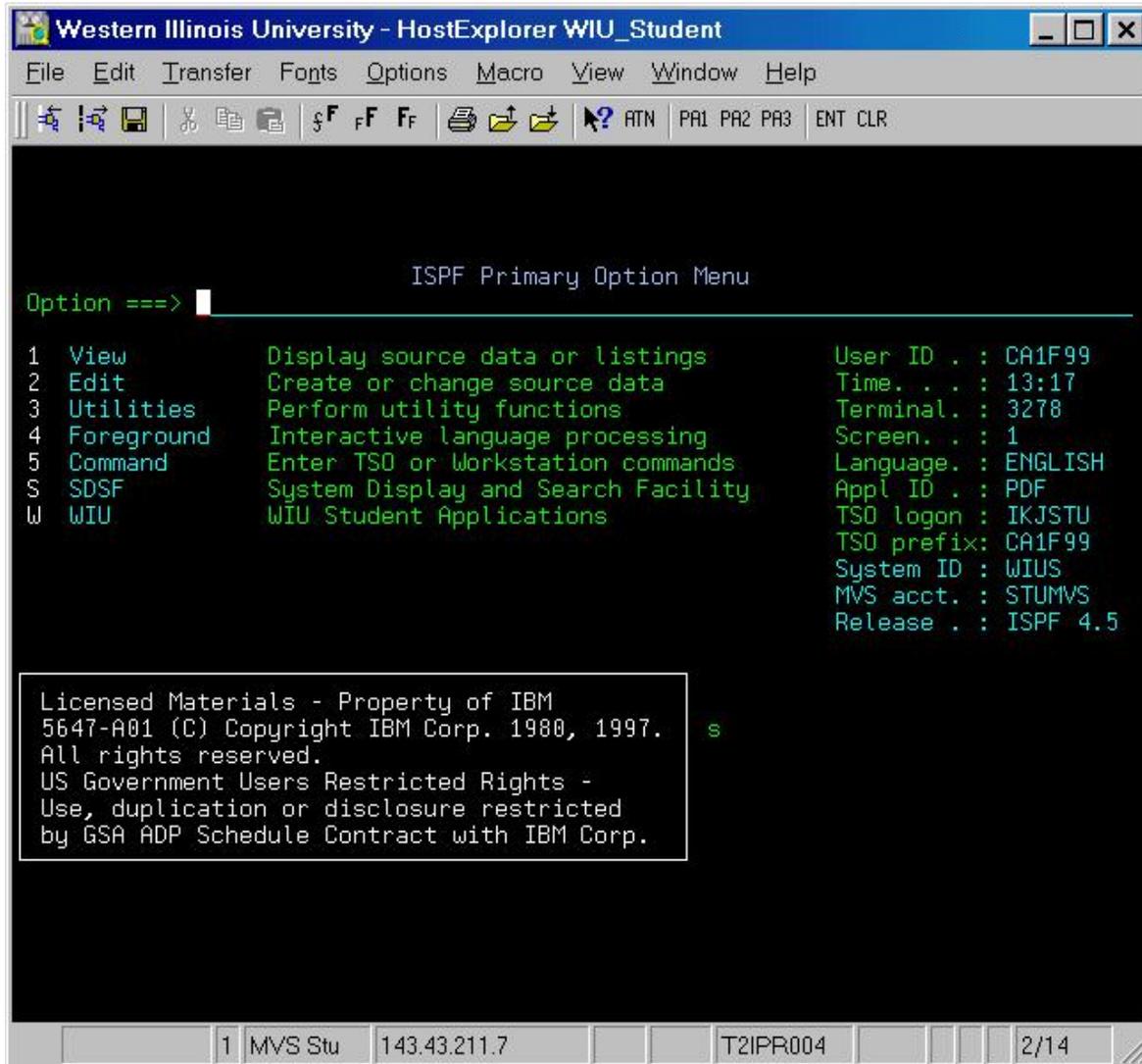


Figura 1.5: El ambiente TSO.



1.4 El monitor de transacciones CICS (Aplicaciones en línea).

Existen dos tipos de datos: Datos maestros y datos transaccionales.

Los datos maestros describen algunos aspectos del mundo. Sin embargo, el mundo cambia. La necesidad de modificar esos datos maestros y ponerlos en concordancia con el mundo nuevamente, es llamada datos transaccionales.

El Customer Information Control System (CICS) es un producto de software que atiende aplicaciones Online en el Mainframe. Actúa como una interfase entre el MVS y sus programas aplicativos. En un entorno de procesamiento Online, se necesitan muchas funciones de control. El CICS cumple estas funciones, simplificando así el trabajo de programador aplicativo. Puesto que estos servicios de administración tienen lugar en un entorno de multitareas o de subtareas, el CICS debe ocuparse también de la administración de la memoria. En términos actuales, el CICS es un monitor de transacciones.

Procesamiento Online.

Los datos en una transacción Online a diferencia de una Batch, son ingresados por un operador vía terminal. Así, no se programa por adelantado el momento de la entrada de los datos. El programa ha dejado de requerir la lectura de los datos, en lugar de ello, ahora son los datos los que piden la ejecución del programa.

Un mensaje ingresado por un operador de terminal requiere con frecuencia una respuesta inmediata. Por ejemplo, si se ingresa una consulta sobre el nivel de stock de un producto en un inventario, se desea saber de inmediato si se dispone de existencia. La transacción en un sistema Online, se atiende de inmediato.

Una transacción Online se distingue por ser un mensaje de entrada y una respuesta de salida. Los mensajes pueden ser transmitidos concurrentemente desde varias terminales. Si aquellos mensajes fueron de tipos diferentes (consultas, altas, actualizaciones) cada uno es procesado por el programa adecuado. Esto significa que en lugar de escribir un programa largo y complejo para procesar diferentes tipos de transacciones, el programador puede escribir varios programas cortos, y por lo tanto relativamente simples, para procesar cada uno de los tipos de transacción. Una transacción o conversación Online actualiza la información en tiempo real para garantizar la integridad de esta.

Características del procesamiento Online con respecto al procesamiento Batch.

En un entorno Online, los departamentos de una empresa pueden ingresar datos concurrentemente y al azar, cada uno de ellos en sus propias terminales. Una transacción ingresada en una terminal consta de un código de transacción, también llamado “ID de transacción” (predefinida por el programador de sistemas) y de datos. Cada una de las transacciones crea una unidad de trabajo dentro de CICS llamada tarea (task).



Varias transacciones pueden acceder a los mismos recursos al mismo tiempo. Los programas requeridos para atender transacciones concurrentes deben estar disponibles para su ejecución concurrente.

Transacción ingresada desde una terminal.

Para su ejecución concurrente, ha de existir una interfase que pueda hacer lo siguiente:

- Aceptar datos tan pronto estén listos para ser transmitidos.
- Recordar de que terminal provienen los datos.
- Llamar al programa adecuado y transferirle los datos.
- Controlar el uso compartido de recursos.

Esta interfase es el CICS.

El CICS supervisa toda la actividad de comunicación de datos, lo que significa que:

- Administra las terminales.
- Administra los datos.
- Administra los programas de aplicación.

Como ya se mencionó, una transacción ingresada por un operador de terminal consta de dos partes, una “ID de transacción” seguida de los “datos” que han de ser procesados. Véase el ejemplo en que un operador desea consultar la existencia del producto “1234” en un inventario con el programa aplicativo definido por la transacción “D138”, realmente se esta enviando una ID de transacción “D138” seguida de los datos “1234”; cuando esto ocurre, un módulo de administración CICS la lee en la región CICS. Los datos se transfieren a un Buffer de terminal para alojar el mensaje.

Se debe mencionar que no puede ingresarse ninguna otra transacción desde la terminal usada hasta que se haya completado la transacción requerida actual. Luego de haber sido ingresada la transacción, el CICS valida la ID de transacción. Toda ID de transacción ha de predefinirse y estar asociada con un programa de aplicación.

Para una ID de transacción válida, el CICS creará una tarea para procesar esa transacción. Esta tarea es procesada por TASK1.

TASK1 es en realidad un bloque de control que contiene información que el programa de aplicación y el CICS necesita para llevar a cabo la unidad de trabajo a completar.

El siguiente paso es cargar, si fuera necesario, el programa de aplicación que ha sido codificado para procesar la transacción D138.

Ahora puede comenzar el programa de aplicación que procesa la transacción D138. Puesto que esta es una transacción de consulta, el programa emite un pedido de lectura al



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

archivo maestro de partes y obtiene el registro cuya clave de producto es “1234”. El registro se transfiere a un buffer de archivo, que luego se asocia con la tarea.

Una vez recuperado el registro, el programa de aplicación da formato a la respuesta de la consulta. Un modulo de administración de CICS transmite este mensaje a la terminal y la transacción termina.

Al terminar la transacción, se liberan los recursos adquiridos y las conexiones establecidas durante el procesamiento de la tarea. La terminal queda luego disponible para el ingreso de otra transacción.

El entorno CICS.

El CICS controla la ejecución concurrente de los programas. Diferentes programas pueden procesar concurrentemente diferentes tipos de datos o el mismo programa puede procesar concurrentemente diferentes mensajes con el mismo código de transacción. Por ejemplo, mientras una transacción esta en estado de espera (esperando quizás que termine una operación de entrada/salida), puede procesarse una segunda transacción desde otra terminal. Así varias transacciones que están siendo atendidas concurrentemente por el mismo programa pueden provocar la creación de diferentes tareas que comparten las mismas instrucciones.

Puesto que varias tareas pueden compartir el mismo programa de aplicación, las instrucciones del programa de aplicación no han de experimentar modificaciones; esto es, durante la ejecución no debe modificarse la codificación.

El CICS es el programa principal de la región en que se encuentra ubicado, los programas de aplicación operan bajo el CICS.

Todo programa de aplicación Online debe devolver el control al CICS cuando termine su ejecución. Los programas no se modificarán durante la ejecución de los mismos.

Componentes del CICS.

El sistema CICS consta de:

- Módulos de Administración.
- Tablas.
- Bloques de control.

Los módulos de control son los programas CICS que mantienen una interfase entre el sistema operativo MVS y los programas de aplicación. Todo módulo de administración cumple con una función particular. Por ejemplo, cuando un programa de aplicación emite un pedido de lectura de un registro, el módulo de administración FILE Control Program procesa el pedido. Cuando un programa de aplicación emite un pedido de transmisión de un mensaje a una terminal, el modulo de administración TERMINAL Control Program actúa como la interfase. El modulo de administración encargado de la ejecución de los programas



es el PROGRAM Control Program. Los pedidos de entrada/salida se hacen al CICS, en lugar de hacerlo al sistema operativo como lo hace el procesamiento Batch.

Las tablas definen el entorno del sistema CICS. Las tablas generadas por el programador de sistemas, están funcionalmente asociadas con los módulos de administración. Por ejemplo, todas las definiciones de archivos están en la File Control Table (FCT) de modo que todos los programas de aplicación y tareas puedan compartirlas. La Terminal Control Table (TCT) define cada terminal de la red; el programa de aplicación no necesita así ocuparse de los atributos físicos de las terminales del sistema. La Program Control Table (PCT) contiene la relación transacción – programa y la Processing Program Table (PPT) contiene la relación de programas aplicativos y mapas, la Resource Control Table (RCT) contiene la relación de planes de base de datos y transacciones, es decir la liga entre DB2 y CICS. No todos los módulos de administración tienen tablas asociadas.

Los bloques de control contienen información sobre el tipo de sistema. Al iniciarse una transacción se crea un bloque de control llamado Área de Control de Tarea (TCA). La TCA contiene información perteneciente a la tarea.

1.5 El tipo de dato Queue en CICS.

En CICS una Transient Data Queue⁹ representa un área de memoria dentro de la región CICS denominada “destino”. Se llama destino debido a que el administrador CICS previamente asigna a esta Queue una salida a la consola (spool) del sistema operativo MVS. El destino es escrito desde un programa aplicativo CICS para que al ser depositados los datos en la Queue, el JES2¹⁰ inmediatamente interprete el contenido como una tarea de procesamiento Batch que necesita ser atendida y ejecutada inmediatamente, es decir, es el destino quien permite ejecutar procesos Batch desde una tarea Online, el JES2 también hace posible ver el resultado de la ejecución del procedimiento en la consola de salida (spooling).

Opciones y parámetros para las Queue:

- FROM (data-area): Especifica los datos que serán escritos a la Queue.
- LENGHT (data-value): Especifica la longitud de los datos que serán escritos.
- QUEUE (name): Especifica el nombre de la Queue de 1-4 caracteres; El destino debe estar definido en el CICS.
- SYSID (systemname): Especifica el nombre del sistema (sólo para sistemas remotos).

⁹ Una Queue es un tipo de dato nativo del Cics, y es en esencia, una cola.

¹⁰ Job Entry Subsystem 2. Se trata de un lector de procesos batch presente en MVS, este disparador cuenta con un controlador de sentencias llamado JECL que le permite incluso programar y determinar el momento de ejecución de la tarea. Cuando el procedimiento es ejecutado, se envía el resultado a la consola del TSO.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

Condiciones de error:

- DISABLE: Ocurre cuando la Queue ha sido deshabilitada.
- INVREQ: Ocurre cuando se quiere escribir una Queue que no esta en modo de escritura.
- LENGHTERR: Ocurre cuando la Queue esta definida con una longitud diferente a la que se manda como parámetro cuando el programa aplicativo del CICS la escribe.
- NOSPACE: Ocurre cuando ya no hay espacio en el área de memoria del destino.
- NOAUTH: Ocurre cuando un recurso de seguridad ha fallado sobre la Queue.
- NOTOPEN: Ocurre cuando el destino esta cerrado.

El siguiente ejemplo corresponde a código COBOL-CICS y muestra como se escribe en una TD Queue:

```
EXEC CICS WRITEQ TD  
  QUEUE('CSML')  
  FROM(MESSAGE)  
  LENGTH(LENG)
```

Figura 1.6: Se puede apreciar la manera en que se define una estructura de datos de tipo Cola en el CICS.

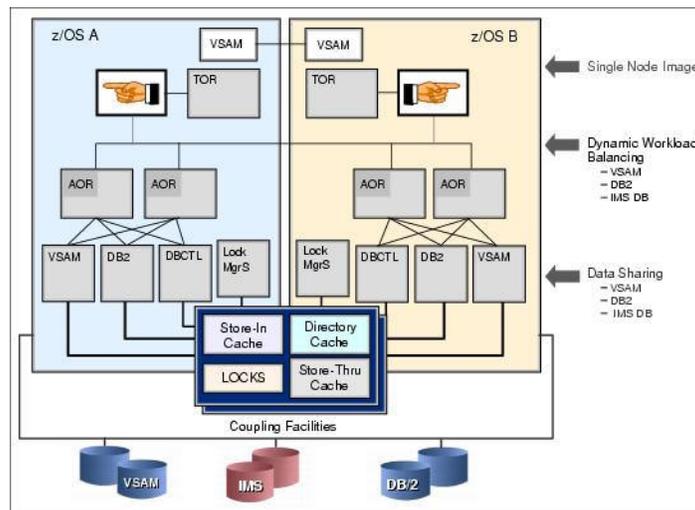


Figura 1.7: Esta figura muestra una típica configuración CICS.



1.6 Bases de Datos con DB2.

El DB2 pertenece al mismo fabricante de la plataforma Mainframe y es obviamente, el manejador de base de datos por excelencia para esta familia de computadoras.

Al hablar de dos maneras de modificar la información, se hace referencia al procesamiento Batch y al procesamiento Online, es por medio del DB2 como estas dos técnicas impactan los datos. Si bien es cierto que es menos costoso que los programas aplicativos interactúen con archivos de trabajo que con bases de datos, también es verdad que el DB2 hace un excelente trabajo relación desempeño-beneficio. Las Bases de Datos son los medios preferidos para la *persistencia de la información*.

Storage Group.

Un STORAGE GROUP DB2, es un conjunto de volúmenes implementados en un dispositivo de almacenamiento de acceso directo (DASD). Estos volúmenes contienen los archivos en los cuales los TABLESPACE o índices son almacenados. El máximo de volúmenes por STORAGE GROUP es de 133.

Todos los volúmenes asignados a un STORAGE GROUP deben tener el mismo tipo de dispositivo, sin embargo, algunas partes de la base de datos pueden estar almacenadas en diferentes STORAGE GROUP.

Database.

Una DATABASE es una colección organizada de objetos relacionales tales como TABLESPACES o INDEXSPACES los cuales son agrupados para facilitar las operaciones y garantizar su seguridad. Cada DATABASE tiene su propio catálogo, archivos de LOG, seguridad y parámetros de performance. El SQL es usado por el DB2 para implementar consultas (queries) y explotar la información que existe en ella. Antes de examinar el SQL es necesario conocer los objetos básicos que son definidos para cada base de datos DB2. Este objeto no tiene existencia física y existen comandos de operaciones sobre él. Por estándar el nombre asignado al DATABASE debe ser alfanumérico de 8 caracteres máximo, en donde el primer carácter debe ser alfabético.

Tablespaces.

Se conoce como TABLESPACE al área de almacenamiento externo que se utiliza para almacenar los registros de una o más tablas. Por estándar el nombre asignado al DATABASE debe ser alfanumérico de 8 caracteres máximo, en donde el primer carácter debe ser alfabético. Un TABLESPACE corresponde a uno o más archivos LDS VSAM¹¹ y es un archivo con un arreglo lineal de páginas de 4K, los TABLESPACE se dividen en

¹¹ Archivos planos con acceso y búsqueda indexada.



unidades del mismo tamaño llamadas páginas. En cada página se almacenarán 1 o varios renglones de una tabla, el máximo es de 117 renglones en una Página. También pueden definirse páginas de 32K, el tamaño máximo de TABLESPACE es de 64gb.

Cuando se crea un TABLESPACE es necesario especificar a que DATABASE pertenece y dentro de que STORAGE GROUP se debe ubicar, de no ser así DB2 asigna el DATABASE y STORAGE GROUP por default. Existen 3 tipos de TABLESPACE: simple, segmentado y particionado. A continuación se detalla cada uno de ellos.

Tablespaces Simples.

El TABLESPACE simple es el default para DB2, este tipo de TABLESPACE puede contener más de una tabla y los renglones que contiene son ordenados de acuerdo a la secuencia en que son insertados, es decir, los renglones de las tablas no se almacenan por separado. Es por ello que para encontrar todos los renglones correspondientes a una tabla puede requerir examinar todo el TABLESPACE (scan TABLESPACE). Así es que si la tabla es borrada sus renglones permanecerán y no estará disponible el espacio hasta que el TABLESPACE sea reorganizado. El tamaño máximo de un archivo en un TABLESPACE simple es de 2gb.

Tablespaces Segmentados.

Un TABLESPACE segmentado está pensado para contener más de una tabla. El nombre de segmentado proviene del hecho de que el TABLESPACE está dividido en grupos de páginas denominados segmentos. Cada segmento tiene la particularidad de contener renglones de sólo una tabla. Para buscar todos los renglones de una tabla no será necesario examinar todo el TABLESPACE, sino que sólo se examinarán los segmentos que contenga esa tabla. Cuando la tabla sea borrada, sus segmentos serán reusables inmediatamente. Un TABLESPACE segmentado puede tener de 1 a 32 archivos lineales VSAM y al igual que el TABLESPACE simple, el tamaño máximo de un archivo es de 2 Gb.

Tablespaces particionados.

El TABLESPACE particionado puede contener sólo una tabla y se recomienda su uso para tablas ‘largas’, es decir, del orden de 2 millones de renglones. En este TABLESPACE el espacio disponible esta dividido dentro de unidades separadas de almacenamiento llamadas particiones. Cada partición tiene su correspondiente archivo VSAM. El tamaño máximo del archivo depende del número de particiones, de tal forma que de 1 a 16 particiones el máximo es de 4Gb, de 17 a 32 particiones un máximo de 2Gb y, de 33 a 64 particiones de 1Gb.



Características de un TABLESPACE particionado.

- Contiene una sola tabla por TABLESPACE.
 - Debe existir un índice clúster asociado a este TABLESPACE, ya que es aquí donde se define el criterio de particionamiento.
 - Las columnas que forman el índice clúster **NO PUEDEN SER ACTUALIZADAS**.
 - El tamaño máximo de los archivos depende del número de particiones.
- Ventajas de un TABLESPACE particionado.
- Cada partición puede ser almacenada en diferente STORAGE GROUP, facilitando de esta manera el control de espacios.
 - Debido a que DB2 pueden trabajar sobre una partición en particular se pueden ejecutar varias utilerías sobre el mismo TABLESPACE al mismo tiempo.
 - DB2 puede decidir resolver una consulta en específico utilizando varias particiones a la vez, a lo que se le llama paralelismo.
 - Se facilita el manejo de utilerías, ya que el espacio de trabajo requerido es dividido en pequeñas porciones, siempre y cuando estas se manejen a nivel partición.

Desventajas de un TABLESPACE particionado.

- El número de particiones definido es fijo, por lo que un re-particionamiento con datos en la tabla implicaría dar DROP al TABLESPACE y volverlo a crear.
- No se pueden actualizar valores de los campos contenidos en el índice cluster.
- La búsqueda de espacio dentro del TABLESPACE no es tan eficiente como en un TABLESPACE segmentado.

Tablas.

Todos los datos en DB2 se presentan a través de tablas. Cada tabla contiene los datos de una entidad específica y está ordenada en columnas. Se puede definir una TABLA como un conjunto de renglones, en donde cada renglón contiene las mismas columnas. La representación almacenada de un renglón es llamada registro, y la representación almacenada de una columna es llamada campo. Todos los datos contenidos en una columna deben ser del mismo tipo. Una tabla puede tener una llave primaria (primary key), una primary key es una columna o un conjunto de columnas cuyos valores identifican de manera única cada renglón. El nombre de la tabla puede constar de hasta 18 caracteres alfanuméricos, el primero debe ser alfabético.

Índices.

Un ÍNDICE es un conjunto ordenado de apuntadores a los datos de una tabla de DB2, es decir, es un medio de acceso directo a los renglones de una tabla. El índice se almacena en forma separada a la tabla. Los índices de DB2 se definen sobre una o varias columnas de



una tabla y depende del valor de los datos de cada columna. Se pueden definir varios índices para una tabla, pero la decisión final de utilizar índices corresponde al optimizar el DB2.

Cada índice ocupa su propio indexspace, se llama indexspace al área de almacenamiento externo usada para almacenar un índice y puede constar de 1 a 64 archivos lineales VSAM, se define automáticamente cuando el índice es creado.

El uso de índices permite optimizar el acceso a los datos y garantizar la unicidad, esto quiere decir, que una tabla con un índice único no puede tener dos renglones con los mismos valores en una columna o en las columnas que forman el índice principal. Es importante observar con mucho cuidado en la definición de varios índices para una tabla, ya que al actualizarla se propaga en los índices también. El máximo de caracteres para definir el nombre del índice es de 18, el primer caracter debe ser alfabético. El índice al igual que el TABLESPACE esta formado por páginas de 4K.

Existen algunas variantes en los índices que se pueden definir: unique, no-unique, particionado y clúster. A continuación se detalla cada uno de éstos.

Unique.

El índice UNIQUE no permite la existencia de valores duplicados y en general se define para la llave primaria de una tabla.

No Unique.

El índice NO-UNIQUE permite valores repetidos en las columnas que lo forman. La regla general, es definir un índice no-unique sobre cada llave foránea de una tabla. Se pueden definir varios índices no-unique para una sola tabla.

Particionados.

Un índice creado sobre una tabla en TABLESPACE particionado es un índice PARTICIONADO y está dividido dentro de varios indexspace. Se define en los TABLESPACE particionados para definir el criterio de particionamiento. Sólo puede haber un índice particionado cuando se tenga un TABLESPACE particionado. El atributo con el que se debe definir es cluster.

Clúster.

Con el índice CLUSTER, DB2 intenta mantener el orden físico de los renglones igual al orden lógico del índice. El índice cluster determina el orden aproximado en el cual los registros de la tabla son almacenados. DB2 puede acceder a toda la tabla más rápido por la llave cluster que con cualquier otra llave. Cada tabla puede tener sólo un índice cluster, se recomienda sobre los índices no-unique. La ventaja que representa la definición de un



índice cluster es que controla la secuencia de los renglones durante un insert o reorg, además de que reduce los costos de una consulta para índices de baja cardinalidad, es decir, con pocos valores diferentes.

Vistas.

Una VISTA es una vía alterna para representar datos que existen en una o más tablas, es decir, es un subconjunto de columnas y/o renglones formados a partir de 1 o varias tablas. La peculiaridad de las vistas que no existen físicamente y su definición es guardada en el catálogo, su nombre puede ser de hasta 18 alfanuméricos y tiene la restricción de que el primer carácter debe ser alfabético. El empleo de las vistas se hace para reforzar la independencia de los datos respecto a los programas que los usan, o bien por razones de seguridad. Las vistas pueden estar basadas en otras vistas o en una combinación de vistas y tablas. No se puede crear un índice sobre una vista, pero los índices creados sobre las tablas base funcionan de igual manera para la vista.

Sinónimos.

Un SINÓNIMO es un nombre alternativo para una tabla o vista. Cualquier usuario puede definir sus propios sinónimos, la longitud máxima para el nombre del sinónimo es de 18 caracteres y el primero debe ser alfabético.

Tipos de Datos en DB2.

Cuando una base de datos es implementada, es necesario asignar a cada columna un tipo de dato y la longitud del campo para ese dato. El primer paso que se debe seguir cuando se define una columna, es que tipo de dato contendrá. Los tipos de datos en DB2 válidos son tipo carácter, numéricos y de fecha. A continuación se presenta el esquema de los tipos de datos para DB2.

Carácter.

Sintaxis.

CHAR (L)

Donde: L = longitud, puede ser de 1 a 254 caracteres.

VARCHAR (L)

Donde: L = longitud máxima, puede ser de 1 hasta el valor del registro con un límite de 32764.



Char.

Los campos definidos como CHAR son almacenados en la base de datos usando la cantidad definida para su almacenamiento.

Sin embargo, el uso de campos tipo char puede potencialmente desperdiciar espacio en disco si el dato no está usando la cantidad definida para su almacenamiento. La longitud de un campo string puede ser de 1 a 254 caracteres, si no se define un valor para la longitud se asume el valor de 1.

Varchar.

Los campos definidos como VARCHAR son almacenados en la base de datos usando únicamente la cantidad de espacio requerido para almacenar los datos. Si un campo tipo varchar es actualizado y el valor resultante es mayor que el original, el registro será movido a otra página en la tabla. La existencia de muchos registros de este tipo puede causar una importante degradación en el desempeño, ya que varias páginas deben ser consultadas para obtener solo un registro. La longitud máxima de una columna varchar es de 4,000 bytes.

Numéricos.

SMALLINT. Su rango de valores es de -32768 a 32767.

INTEGER. El rango de valores va de -2,147,483,648 a 2,147,483,648.

DECIMAL(X,Y) El máximo valor permitido para X es 31.

Donde X = Número total de dígitos

Y = Número de dígitos decimales.

FLOAT(N)

Donde N = Longitud del tipo de dato para cantidades enormes (precisión sencilla) o cantidades muy pequeñas (precisión doble).

Existen varios tipos de datos que pueden ser usados para almacenar datos numéricos y sus precisiones. La precisión de un número es el número de dígitos usados para representar su valor. A continuación se detallan los tipos de datos numéricos para DB2.

Integer.

Los campos definidos como INTEGER ocupan el doble de espacio que los tipos smallint, pero tiene un mayor rango posible de valores. El rango de valores va de -2, 147, 483, 648 a 2, 147, 483, 648. La precisión de un entero es de 10 dígitos a la izquierda del decimal. Se necesitan 4 bytes para el almacenamiento de las columnas definidas como tipo integer. Un small integer SMALLINT usa la menor cantidad de almacenamiento en una base de datos y no permite ningún dígito a la derecha del decimal. Su rango de valores es de



-32768 a 32767. La precisión de un smallint es de 5 dígitos (a la izquierda del decimal). Para su almacenamiento requiere de 2 bytes.

Decimal.

Los campos definidos como DECIMAL son usados para números con fracciones, este tipo de datos son almacenados en un formato empacado conocido como notación decimal código binario (BCD). La precisión es el total de número de dígitos (rango de 1 a 31) y la escala es el número de dígitos en la parte fraccionaria del número. Los términos NUMERIC, NUM, DECIMAL y DEC pueden ser utilizados para declarar una columna con formato numérico decimal. Si el valor de la precisión y la escala no son declarados en las columnas, el valor default es de (5,0).

Punto flotante.

Un tipo de dato REAL es una aproximación de un número. La aproximación requiere de 64 bits u 8 bytes de almacenamiento.

La longitud de un tipo de dato de precisión sencilla usa el tipo de dato FLOAT y debe ser definido entre 1 y 24. El rango de valores para un tipo de dato real es $-3402E+38$ a $3402E+38$.

Para especificar un número de doble precisión se debe usar el tipo de dato FLOAT cuya longitud debe estar definida entre 25 y 53. El FLOAT requiere para su almacenamiento 8 bytes (float de doble precisión) y el REAL 4 bytes (float de precisión sencilla).

Fechas.

- DATE:

Formato interno: YYYYMMDD

Formato externo (ISO): YYYY-MM-DD

- TIME:

Formato interno: HHMMSS

Formato externo (ISO): HH.MM.SS

- TIMESTAMP:

Formato interno: YYYYMMDDHHMMSSNNNNNN

Formato externo (ISO): YYYY-MM-DD-HH.MM.SS.NNNNNN

Los tipos de datos fecha y hora pueden ser tratados como caracteres o datos tipo string y no son almacenados en la base de datos como caracteres de longitud fija.



Date.

Utiliza 4 bytes para su almacenamiento, externamente tiene una longitud de 10 bytes. Su representación dependerá del código del país y el formato en que se use. Existe una opción en el BIND llamada DATETIME en donde se define el formato externo de los valores de la fecha y la hora.

Time.

Utiliza 3 bytes para su almacenamiento, externamente tiene una longitud de 8 bytes. Su representación externa puede ser cambiada utilizando el BIND.

Timestamp.

Este tipo de dato tiene un formato externo sencillo, pero la opción del bind datetime no afecta el formato del timestamp. La representación externa del datetime es en año-mes-día-hora- minuto-segundo- microsegundo. Cada columna definida como timestamp requiere 10 bytes para su almacenamiento interno.

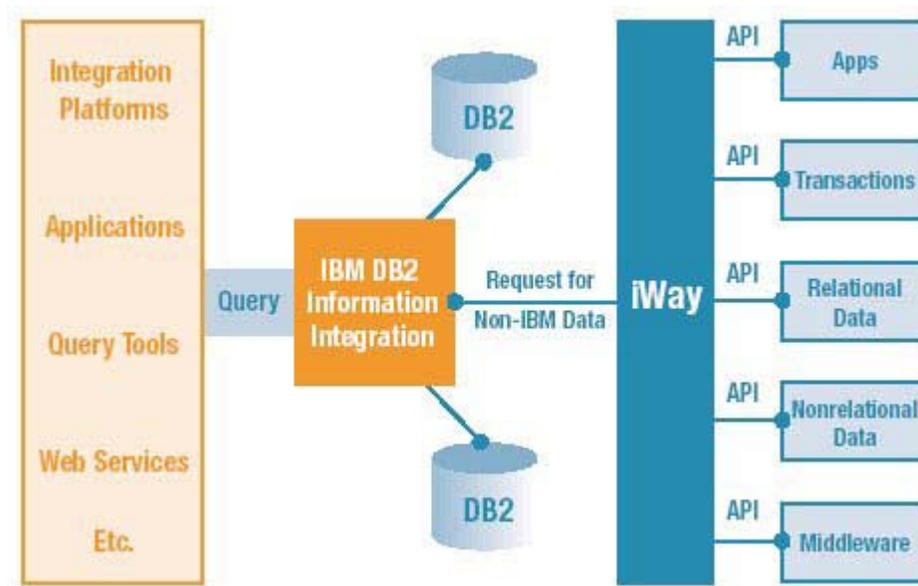


Figura 1.8: Papel del DB2 dentro de la plataforma Maiframe.



Interfaz DB2 con CICS.

RCT

La conexión de DB2 con CICS se define e implementa a través de RCT (Resource Control Table, Tabla de Control de Recursos). Esta tabla describe las relaciones entre las transacciones de CICS y los recursos de DB2. Las facilidades con que cuenta DB2 contienen una macro para generar el RCT.

Utilizando las tareas macro de CICS, se puede definir lo siguiente:

- RCT entradas para una transacción o un grupo de transacciones y sus respectivos identificadores.
- Nombres de Plan.
- Opciones de chequeo de autorización.

La RCT debe ser linkeditada dentro de una librería que sea accesible a MVS, no se debe dar link a RCT como reentrante o reusable porque esto causaría que en los pasos del CICS se originara un '*abend*'¹² al inicio.

La conexión entre DB2 y CICS requiere que se generen varias tablas de CICS con entradas adicionales. Se aplica una macro con DB2 para definir las conexiones entre CICS y DB2 usando la tabla de control de recursos RCT.

Interfaz con DB2.

Paquetes.

Un paquete se produce a partir de un solo DBRM y es el resultado de dar BIND PACKAGE. Se conoce también como un objeto que contiene un conjunto de declaraciones SQL que han sido ligados y están disponibles para su procesamiento.

Existe la opción de hacer bind a cada paquete:

```
BIND PACKAGE(APLIC1) MEMBER(PROGA)  
ENABLE(CICS) OWNER(GRP1) VALIDATE(BIND)  
RELEASE(COMMIT) ISOLATION(CS)
```

Parámetros del comando BIND PACKAGE

APLIC1

Es el nombre de una colección que agrupa a un conjunto de paquetes.

¹² Error fatal en aplicación.



MEMBER

Es el nombre del programa que se le hace BIND.

ENABLE

Especifica el ambiente en que puede ejecutarse el paquete: CICS, batch, IMS, etc.

OWNER

Es el 'autorization ID' propietario del paquete. El ID debe tener autoridad para ejecutar todos los postulados de SQL que están dentro del paquete.

QUALIFIER

Es el calificador a usar en nombre de tablas y vistas que no tengan calificador explícito. Si no se especifica, se usará el valor de OWNER.

Planes.

- Un plan se genera con el BIND PLAN/PACKAGE a partir de varios paquetes y/o DBRMs.

Colecciones.

Una colección es una etiqueta cualquiera de hasta 18 caracteres que no tiene existencia física, simplemente identifica a un conjunto de paquetes.

Integridad Referencial.

La INTEGRIDAD REFERENCIAL es la ejecución de todas las referencias forzadas. Contar con integridad referencial no significa que los datos están necesariamente correctos.

Se denomina referencia forzada a las limitaciones de un conjunto de valores de una llave foránea, con un conjunto de valores de una llave primaria.

DB2 y/o Aplicativa.

Se puede implementar la integridad referencial por DB2, o se puede implementar a través de programas. En cualquier caso, se debe considerar como es afectada una tabla por sus tablas padre o quizá por sus tablas dependientes. DB2 no fuerza automáticamente la implementación de integridad referencial a través del sistema.

Constraints.

DB2 automáticamente ejecuta un forzamiento (constraint) si la llave primaria y secundaria son definidas explícitamente. Por ejemplo, los objetos con referencias forzadas (referential constraints) deben ser definidas algunas veces en cierto orden. Una llave foránea no puede ser definida sin que su correspondiente llave primaria exista y tiene un índice unique (es decir, el índice primario) definido.



Las consideraciones de recuperación son más complejas para objetos con referencia forzada (referential constraint). DB2 automáticamente fuerza la integridad referencial durante las operaciones SQL insert, update y delete. Pero esto no previene en lo absoluto la consistencia en las tablas, ya que una operación produce bastantes mensajes warning, entonces la administración total de la operación de recuperación es responsabilidad del usuario.

En resumen, la ejecución automática de la integridad referencial es una poderosa herramienta, pero su uso requiere de una cuidadosa planeación y atención.

Primary y Foreign Key (Llave primaria y secundaria).

Una tabla puede tener sólo un índice primario, y tiene las mismas restricciones que una llave:

- Puede incluir hasta 64 columnas.
- La columna no puede ser nombrada dos veces.
- La suma de la longitud de la columna no puede ser mayor que 254.
- Todos los valores de una llave primaria son únicos y no permite valores nulos (es NOT NULL).

Las reglas de las llaves foráneas son:

- Crear una tabla con una llave foránea requiere más que el privilegio usual sobre las tablas dependientes, también requiere el privilegio del alter para la tabla padre.
- Las descripciones de las columnas de las llaves foráneas deben coincidir con las descripciones de las columnas de las llaves primarias a que se hace referencia, en el mismo orden. Es importante mencionar que los nombres de las columnas de las llaves foráneas no necesitan tener el mismo nombre que las columnas de las llaves primarias y sus atributos nulos y valores de default pueden diferir de las columnas de la llave primaria. Si un índice es definido en las columnas de las llaves foráneas, las columnas del índice pueden ser ascendentes o descendentes y pueden ser diferentes del atributo ascendente o descendente de su correspondiente índice primario.
- La tabla dependiente debe existir y tener una definición completa. La dependiente no puede ser una tabla en el catálogo de DB2 y no tiene que estar en la misma base de datos DB2 o en el mismo TABLESPACE.
- Una llave foránea no puede hacer referencia a una vista.



Reglas de Borrado.

On delete cascade.

- Hacer cascada y borrar la entidad primaria y las dependientes

Set Nulls.

- Nulificar todas las llaves foráneas de las entidades dependientes
- Si se elige esta regla, al menos una columna de la llave foránea debe permitir nullos.

On delete restrict.

- Restringir el borrado de una entidad padre si tiene dependientes
- Está regla es el default, la frase ON DELETE RESTRICT puede ser omitida.

Existen utilerías de DB2 que pueden ser utilizadas en la ejecución de un JCL (procedimiento Batch), algunas son: cargas, descargas, y borrado de datos a partir de tablas de DB2. La utilería se incluye en el paso de JCL donde se desea ejecutar.

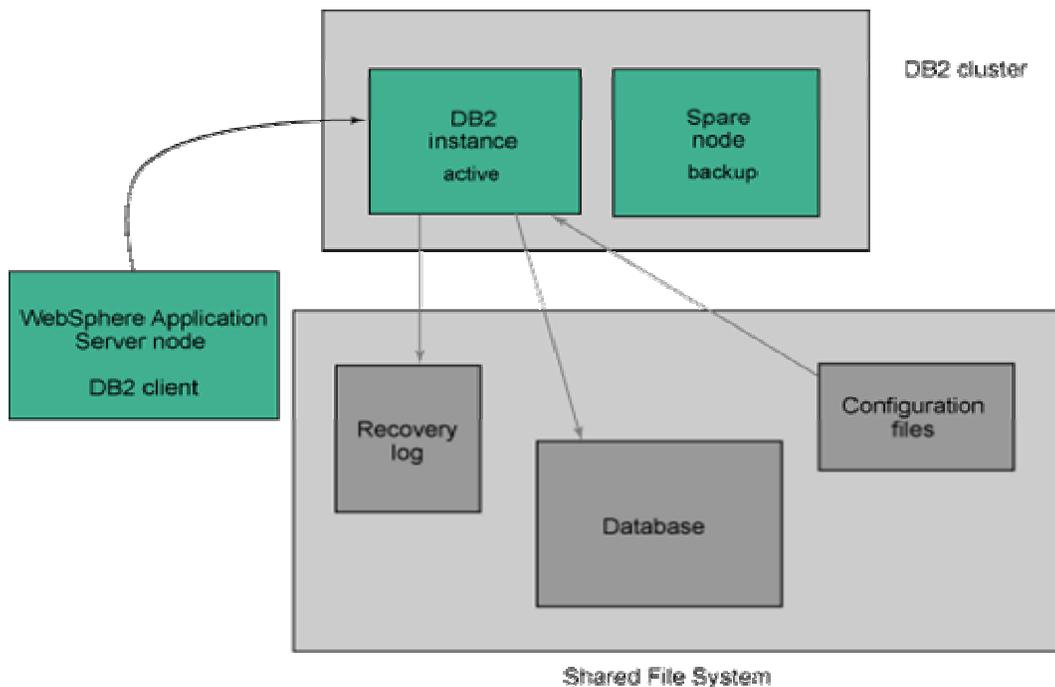


Figura 1.9: Representación de una instancia de Base de Datos DB2..



1.7 El lenguaje de programación COBOL CICS.

Entre los programas aplicativos y el CICS se genera una interfase a nivel comando. Los programas ya no necesitan llamar directamente a varios módulos de administración del CICS. Los pedidos de servicio se expresan con comandos dirigidos a la interfase a nivel comando, la que a su vez asegura la comunicación con el modulo pertinente de administración del CICS.

Fases de un programa CICS.

Un programa de aplicación que se ha de ejecutar en la región de CICS ha de ser reentrante cuando sea un programa conversacional.

En general un programa que se ejecuta en CICS puede ser compartido concurrentemente. Así, mientras procesa una transacción, un programa puede atender otra con la misma ID que se inicie en otra terminal. La ventaja en un programa reentrante radica en que sólo es necesario cargar una vez el programa de aplicación, pudiendo luego ser utilizado por múltiples tareas.

El CICS provee a cada tarea que ejecuta el programa una copia del área de trabajo de éste. Esto se debe al hecho de que el contenido del área de trabajo variara entre una tarea y la siguiente.

Todas las áreas adquiridas y liberadas dinámicamente durante la ejecución de una tarea son parte del área de almacenamiento dinámico (DSA) del CICS. Así, la copia del área de trabajo de un programa para una tarea se coloca en la DSA.

Formato de sentencias y comandos CICS.

El COBOL CICS se distingue del COBOL convencional sólo por el uso de sentencias CICS, el formato general de una sentencia CICS es EXECUTE CICS (EXEC CICS) seguido por el nombre del comando y posiblemente por una o más opciones. Se codifica dentro de las columnas 12 a 72.

Sintaxis:

EXEC CICS

Command option (Arg)

END-EXEC.

Donde:

Command: Describe la operación requerida.

Option: Describe facilidades disponibles con cada comando.

Arg: Describe los argumentos con que se ejecuta el comando.



Valores Argumento en COBOL.

Un “Valor-Datos” puede ser una variable COBOL o una constante. El tipo de dato debe ser especificado como uno de los siguientes:

Media palabra binaria	PIC S9(04) COMP.
Palabra binaria	PIC S9(08) COMP.
String	PIC X(n).

Un “Área-Datos” es un nombre de variable representando una estructura de datos.

Las celdas que sirven para direccionar áreas externas al programa son llamadas BLL “Base linkage linkator” por sus siglas en ingles.

Un “Valor-pointer” es el nombre de una celda BLL o un campo que contiene el nombre de una celda BLL.

Un “Nombre” es una literal o un área de datos representando una literal (debe ser de la longitud de la literal).

Un “Filename” es usado en **FILE (Filename)**, especifica el nombre de un archivo. Su longitud es de 1-8 caracteres. Caracteres válidos A-Z, 0-9, \$, @ y # (las letras minúsculas son convertidas a mayúsculas).

Un “Systemname”, es usado en **SYSID (systemname)**, especifica el nombre del sistema. Su longitud es de 1-4 caracteres. Caracteres validos A-Z, 0-9, \$, @ y # (las letras minúsculas son convertidas a mayúsculas).

Una “Etiqueta”, es cualquier nombre de sección en COBOL.

Una hora debe ser una constante o un nombre de variable con el siguiente formato **HHMMSS** donde:

HH representa horas desde 00-24.

MM representa minutos desde 00-59.

SS representa segundos desde 00-59.

Pasos de la compilación de un programa CICS.

Antes de ser compilado, el programa es procesado por un traductor de comandos que convierte los comandos CICS a sentencias MOVE y CALL del cobol.

Durante la ejecución del programa, estos comandos CALL llaman a la rutina de interfase a nivel comando y le transfieren los pertinentes parámetros. La rutina de interfase llama luego al módulo de administración del CICS que corresponde a la función del comando en el programa de aplicación. La siguiente figura ilustra los pasos que han de cumplirse para crear un modulo de carga en la biblioteca de carga.

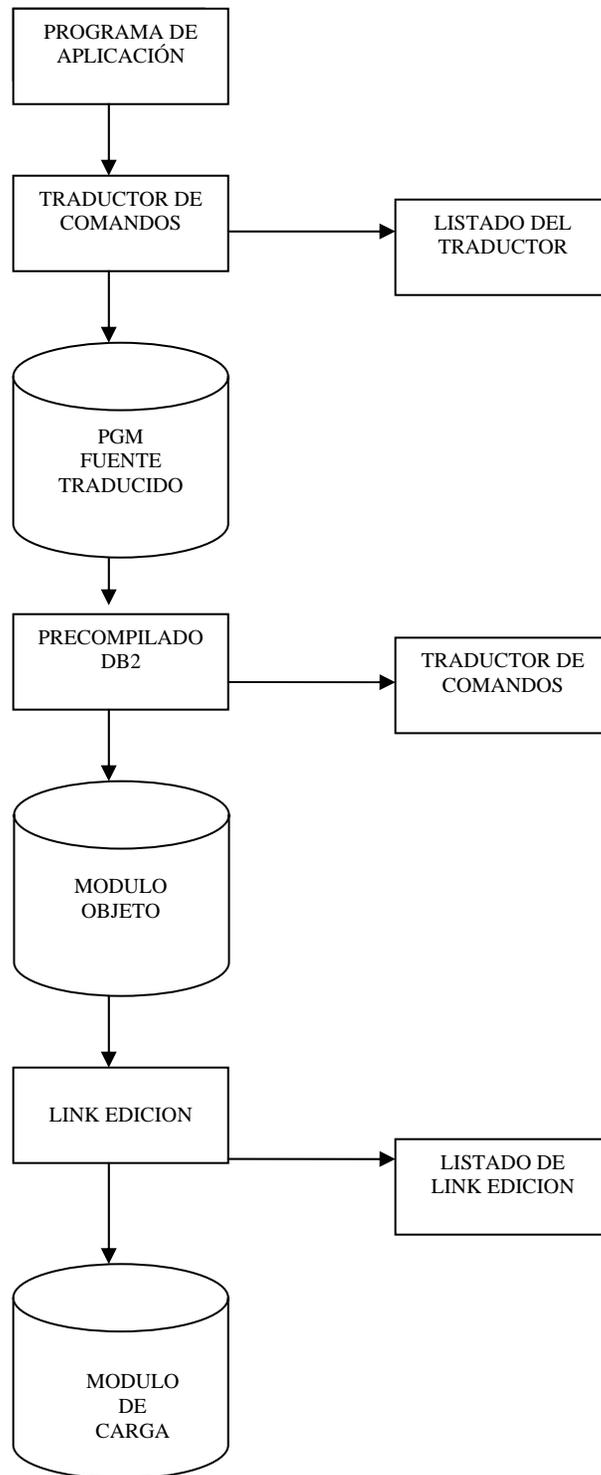


Figura 1.10: Generación del código objeto en la biblioteca de carga.



Bloque de Interfase de ejecución (EIB)

Son datos que el CICS conserva y que el programa de aplicación puede necesitar durante la ejecución.

El CICS copia los datos en un bloque de interfase de ejecución (EIB), que se crea en el momento de iniciarse la tarea, existe un EIB por tarea.

El traductor de comandos incluye automáticamente una copia de EIB en el programa de aplicación.

Un programa aplicativo puede leer todos los campos de EIB, pero no debe cambiar el contenido directamente de ninguno de ellos.

Algunos de los campos más utilizados son **EIBAID, EIBCALEN, EIBDATE, EIBTASKN, EIBTIME, EIBTRMID, EIBTRNID.**

Además de los servicios (tales como lectura de un registro) proporcionados por los varios módulos de administración del CICS, conserva ciertos datos que el programa de aplicación puede necesitar durante la ejecución.

El CICS registra este tipo de datos internos en áreas a las que el programa de aplicación no tiene acceso.

El EIB de una tarea almacena información que el programa de aplicación puede recuperar luego utilizando los nombres de campos EIB.

Condiciones de excepción.

Una condición de excepción representa un error u otra situación anormal que se ha producido durante la ejecución de un comando CICS.

El programador debe anticiparse a todas las condiciones potenciales de excepción.

El CICS cuenta con dos comandos, llamados HANDLE e IGNORE que permiten especificar la acción a tomar en caso de una situación anormal.

Si el programador no especifica lo que debería hacerse el CICS se encargará del error mediante un procedimiento por default, lo cual concluiría en una terminación de ABEND de la tarea.

Por ejemplo, tiene lugar una condición de excepción cuando en el archivo designado no puede hallarse un registro pedido. Otras condiciones de excepción que podrían producirse durante el acceso a un archivo son las siguientes:

- El CICS no reconoce el nombre del archivo.
- El archivo no está abierto.
- Agregar un registro cuya clave ya existe en el archivo.
- No existe la clave especificada.

El programador puede proporcionar acciones para algunas condiciones de error, en el caso en que puedan surgir, para otras, el programador puede aceptar una acción asumida



por default. Cuando una condición anormal ocurre y no es tomada ninguna acción, puede ser codificado un comando IGNORE o una opción NOHANDLE.

Sintaxis:

EXEC CICS HANDLE CONDITION

OPTION (ARGUMENT)

OPTION (ARGUMENT)

END-EXEC

Donde:

OPTION: Palabra clave que identifica la condición de excepción en una sentencia, pueden especificarse hasta 16 condiciones, en cualquier orden.

Ejemplo:

DUPREC Llave duplicada.

NOTFND Registro no hallado.

ARGUMENT: Nombre de la rutina o sección de un programa que se hará cargo de la excepción especificada.

Consideraciones sobre programación.

En cualquier programa se debe usar una rutina de excepción para cualquier situación que provoque una condición de este tipo. Por lo tanto se deben cubrir todas las posibilidades en el comando HANDLE.

El comando HANDLE esta activo desde el momento en que se ejecuta hasta que:

- Termina la tarea.
- Se encuentra otro HANDLE para esa condición.
- Se encuentra un IGNORE para esa condición.

Para dejar sin efecto el HANDLE para una condición, simplemente especificar la condición en la sentencia del HANDLE omitiendo el argumento de la opción o se codifica un IGNORE para esa condición.

Algunas porciones de un programa de aplicación pueden requerir un procesamiento de errores diferentes para la misma condición. En este caso sólo se necesita escribir otro comando HANDLE y colocarlo inmediatamente delante de la sección a la que corresponde. El segundo HANDLE reemplaza al anterior.



Soporte básico de Mapas (Pantallas, interfaz de usuario).

La facilidad del soporte básico de Mapas (BMS) del CICS simplifica la tarea de programar la comunicación entre las terminales y los usuarios. Ventajas:

- BMS independiza al programa de aplicación del:
 - Tipo de terminales usadas (Independencia de dispositivo).
 - Formato de representación de los datos.
- BMS separa el diseño y la generación de los formatos, de la lógica de la aplicación, reduciendo el impacto de uno sobre otro.

Las ventajas anteriores facilitan la construcción de nuevos programas y el mantenimiento al código existente.

El BMS actúa como una interfase entre los programas aplicativos y el programa control de terminales (TCP), que es la responsable de la comunicación física con las terminales.

Independencia de dispositivos.

El programa de control de Terminal (TCP) hace lo siguiente:

- Atiende la transferencia de los datos entre las terminales y el programa de aplicación.
- Busca en la tabla de control de terminales (TCT) una entrada que describa cada una de las terminales.
- Requiere la creación de una tarea para la transacción transmitida desde la terminal.
- Usa las tareas de trabajo definidas en su programa para los datos recibidos desde la terminal o transmitidos a la terminal.

El BMS permite construir programas de aplicación sin ocuparse de las características físicas de las terminales de la red. El BMS prepara el mensaje a transmitir insertando los caracteres de control basándose en el tipo de terminal que se este utilizando.

Independencia de formato.

El BMS también permite el hecho de que no importe la ubicación física de los datos al desplegarse. El programa aplicativo usará nombres simbólicos para referenciar a los campos.

El BMS relaciona el rótulo con la posición real de la pantalla por medio de mapas físicos y mapas simbólicos de descripción.



Mapas físicos.

Un mapa físico describe el formato de la representación de un tipo de terminal dada. Adopta la forma de una tabla que contiene la siguiente información:

- Longitud y ubicación de los campos dados.
- Atributos de los campos dados (Ejemplo: Campos oscuros o campos brillantes).
- Constantes (encabezados o información descriptiva).
- Características de las terminales.

Un mapa físico puede consistir totalmente de constantes, tal como un mensaje al operador.

En la entrada, el BMS elimina todos los caracteres de control y constantes y coloca sólo los datos de trabajo en la memoria del programa de aplicación.

En la salida, el área que contiene los datos se define mediante un mapa simbólico.

Conjunto de Mapas (MAPSET).

Un mapset es uno o varios mapas que se definen para un tipo de Terminal.

Usualmente los mapas se agrupan porque una aplicación o aplicaciones afines utilizan los mapas del conjunto.

Cuando se necesita un mapa en particular, el programa debe identificar el conjunto de mapas que lo incluye.

Mapas Simbólicos.

Los mapas simbólicos definen todos los campos de datos variables.

El mapa simbólico se copia en la memoria de trabajo de un programa, todos los campos definidos están a disposición del programa por su nombre.

Para direccionar cada uno de los ítems del área de trabajo se crea un conjunto de nombres simbólicos para los campos a acceder. El BMS crea el siguiente conjunto de nombres simbólicos:



Nombre Simbólico	Descripción
*NomcampL	El cursor se encuentra posicionado en el campo.
*NomcampA	Byte de atributo del campo. Este byte especifica características tales como la intensidad o color con la cual ha de representarse el campo.
*NomcampF	Byte de señal del campo. Redefine al campo anterior.
*NomcampI	Campo de datos de entrada.
*NomcampO	Campos de datos de salida.

Así los datos de entrada o de salida están a disposición del programa de aplicación en el campo FLD10. El byte de atributo esta disponible en el campo FLD1A y el programa de aplicación puede modificarlo antes de una operación de entrada. Se posiciona el cursor en el campo con FLD1L.

El conjunto de nombres simbólicos integra al mapa simbólico. Uno o más mapas simbólicos forman un conjunto de mapas (MAPSET).

Cuando un programa de Mapa es modificado, después de regenerarse el mismo, se debe recompilar el programa aplicativo que lo usa.

Comando CICS de formateo de entrada.

El CICS es capaz de recibir los datos introducidos en la Terminal mediante un mapa para el proceso de los mismos. La sintaxis necesaria en el programa CICS para realizar lo anterior es la siguiente:

Sintaxis:

EXEC CICS

RECEIVE MAP (nombre)
MAPSET (nombre)
INTO (área de datos) \ SET (referencia-puntero)
TERMINAL {ASIS} \ FROM (área de datos)
LENGTH (valor de datos)

END-EXEC.



Comando CICS de formateo de la salida.

Para los datos que han de formatearse desde el programa de aplicación a la Terminal, el programa emite un comando de formateo de salida.

El BMS aplica los pertinentes caracteres de control al comienzo de control de cada uno de los campos de datos. La corriente de datos resultante a una TIOA, que luego el TCP transmite a la terminal.

Sintáxis:

EXEC CICS

SEND MAP (nombre)
MAPSET (nombre)
TERMINAL
FROM (área de datos) \ SET (referencia-puntero)
LENGTH (valor de datos)
DATAONLY \ MAPONLY
ALARM
CURSOR (valor de datos)
ERASE \ ERASE UP
FREEKB
FRSET

END-EXEC.

1.8 La tecnología de transmisión NDM (Connect: Direct).

Connect: Direct NDM (Network Data Mover) es un paquete de software para transferir datos que opera bajo TCP/IP, distribuye la información y administra las tareas productivas a través de múltiples Mainframes, mini computadoras, estaciones de trabajo y computadoras personales en diversos ambientes y sistemas operativos.

La ventaja que presenta la transmisión vía NDM sobre FTP no sólo es en cuanto a desempeño, también lo hace atractivo el hecho de contar con herramientas de planificación y programación de tareas.

Connect: Direct permite a los centros de procesamiento de datos que están comunicados entre sí por nodos de red:

- Transmitir grandes cantidades de datos.
- Compartir información.
- Planificar transmisiones automáticas por utilería o por medio de aplicaciones (JCL's).
- Controlar y auditar la distribución de información.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

- Transmisión de datos en diferentes plataformas con una interfaz común para todas.

El NDM tiene dos componentes primarios:

Utilería de transmisión de Datos (DTF por sus siglas en ingles): Ejecuta comandos de usuario y transferencia de datos.

Utilería de solicitud de datos Sysplex (SRF por sus siglas en ingles): Implementa la interfaz de Usuario y aplicación (API) al DTF.

La configuración del NDM en un sistema Mainframe con MVS es la descrita por el siguiente diagrama:

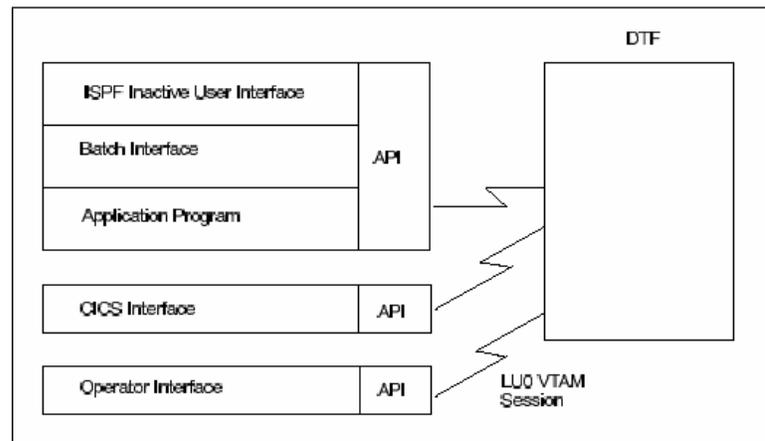


Figura 1.11: Configuración del NDM en un sistema Mainframe.

Se puede observar en el diagrama anterior que todos los componentes están contenidos en módulos residentes en una sola biblioteca de carga (LOADLIB), esta configuración aplica en una máquina con partición o simple.



Interfaz con el CICS.

La siguiente figura muestra la interfaz entre el NDM y el CICS

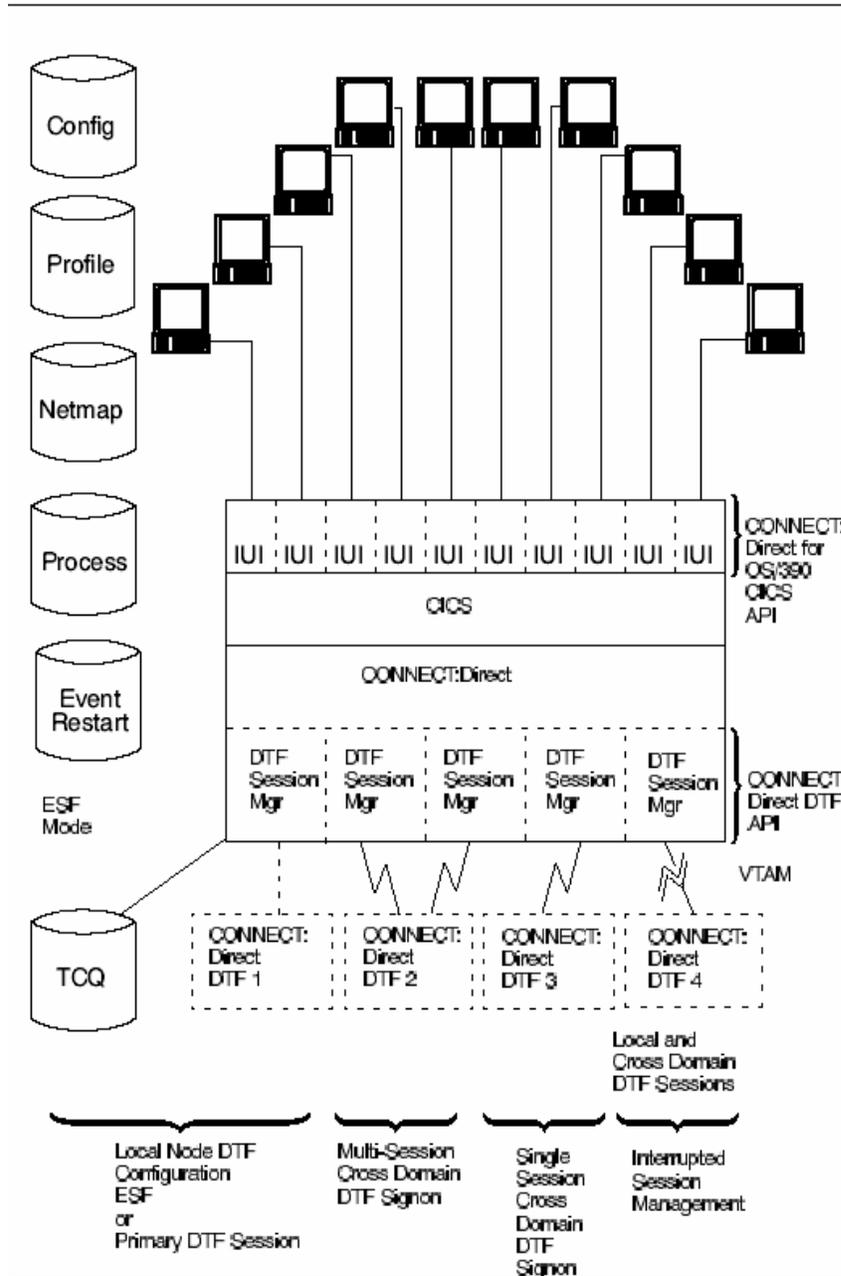


Figura 1.12: El NDM y el CICS.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

A continuación se ilustra como es la transferencia entre dos sistemas Mainframe usando NDM:

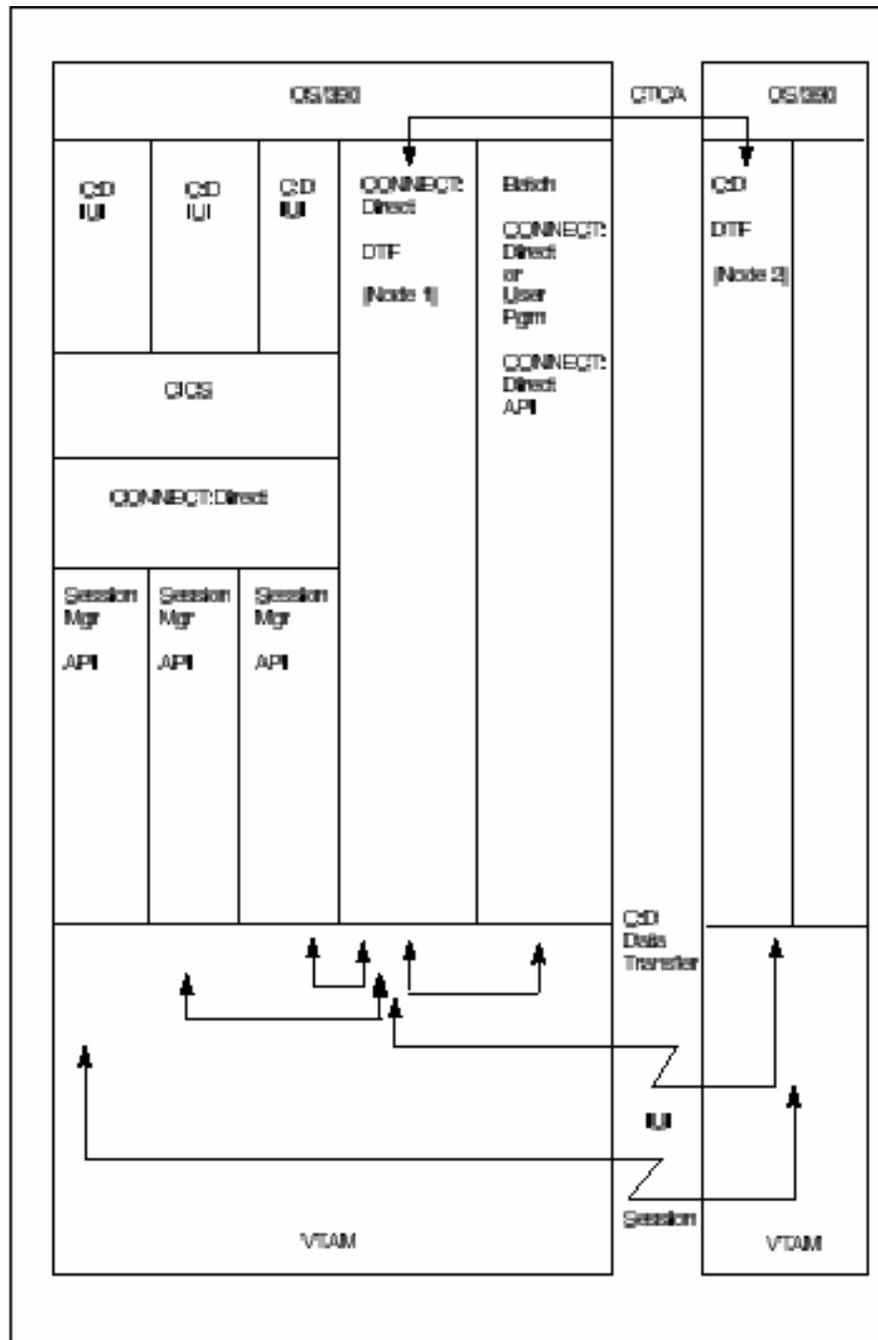


Figura 1.13: Interfaz entre en NDM y el CICS.



El **COBOL** sigue siendo el lenguaje por excelencia para el acceso a Bases de Datos. Las instituciones financieras por el volumen de datos que manejan necesitan hacer uso de estas herramientas.

El **DB2** es un manejador de Bases de Datos relacional especialmente reconocido por su desempeño en el manejo de gran volumen de información; es bien sabido que la correcta combinación de estos elementos, proporcionan aplicaciones robustas de gran calidad y eficiencia.

El **CICS** proporciona el ambiente idóneo para que la banca o cualquier otro grupo financiero puedan realizar gran cantidad de transacciones al mismo tiempo sin sufrir “caídas de sistema” ni pérdida de información.

El **TCP/IP** es el protocolo de transmisión de datos que la mayoría de las plataformas han adoptado (Mac, Windows, Unix, etc.) por lo que el NDM se torna una herramienta simplemente, imprescindible.

1.9 El modelo de redes OSI y el protocolo TCP/IP.

En 1984, la Organización Internacional de Estandarización (ISO) desarrolló un modelo llamado **OSI (Open Systems Interconexión, Interconexión de sistemas abiertos)**. El cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Este modelo es el mejor conocido y el más usado para describir los entornos de red.

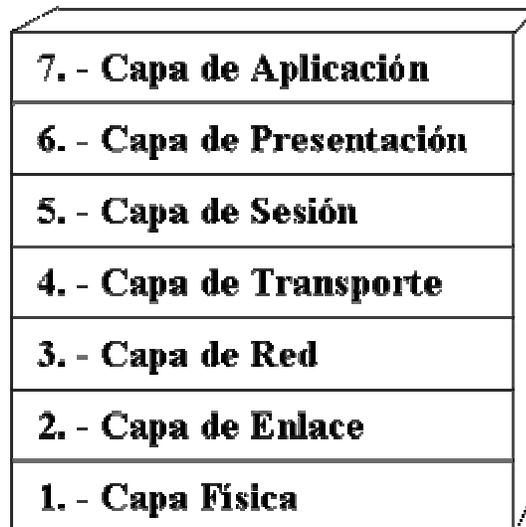


Figura 1.14: Capas del Modelo OSI.



Como se muestra en la figura 1.14, las capas OSI están numeradas de abajo hacia arriba. Las funciones más básicas, como el poner los bits de datos en el cable de la red están en la parte de abajo, mientras las funciones que atienden los detalles de las aplicaciones del usuario están arriba. En el modelo OSI el propósito de cada capa es proveer los servicios para la siguiente capa superior, resguardando la capa de los detalles de como los servicios son implementados realmente. Las capas son abstraídas de tal manera que cada capa cree que se está comunicando con la capa asociada en la otra computadora, cuando realmente cada capa se comunica sólo con las capas adyacentes de la misma computadora.

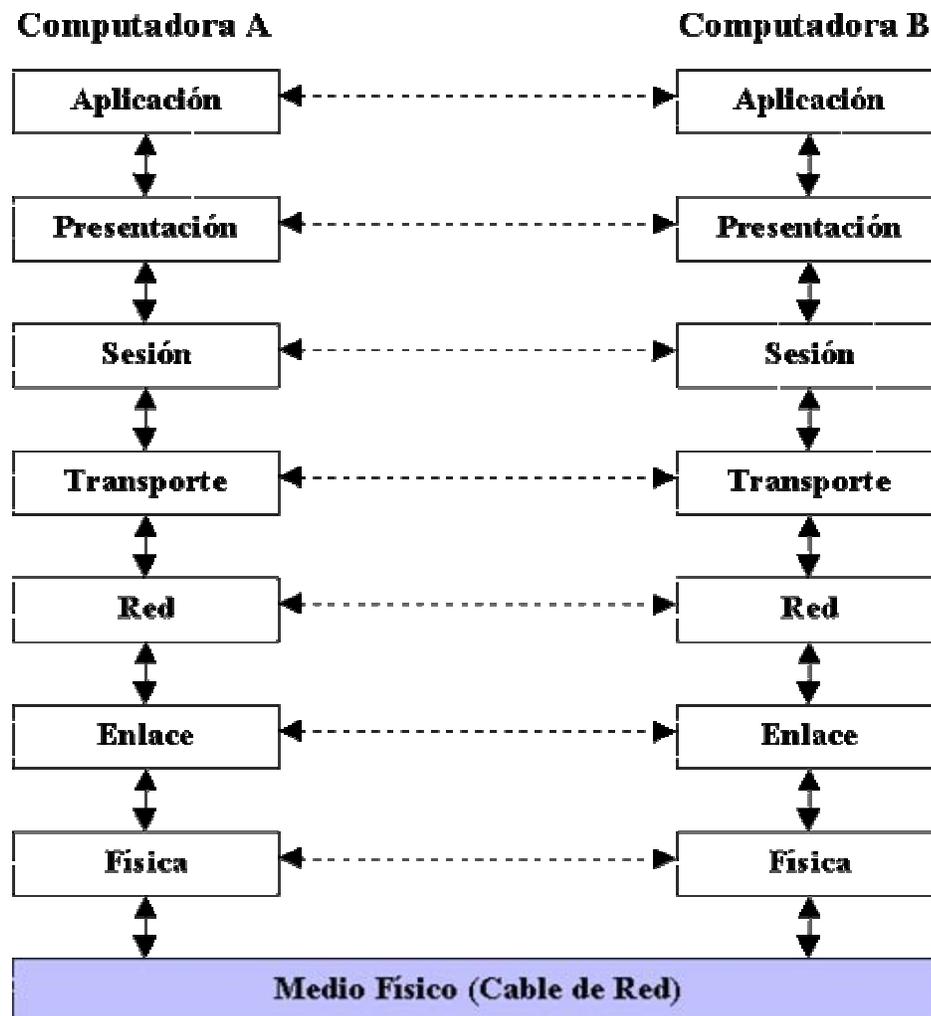


Figura 1.15: Comunicación entre dos computadoras con el Modelo OSI.



Con la figura 1.15 se puede apreciar que a excepción de la capa más baja del modelo OSI, ninguna capa puede pasar información directamente a su contraparte en la otra computadora. La información que envía una computadora debe de pasar por todas las capas inferiores, La información entonces se mueve a través del cable de red hacia la computadora que recibe y hacia arriba a través de las capas de esta misma computadora hasta que llega al mismo nivel de la capa que envió la información. Por ejemplo, si la capa de red envía información desde la computadora A, esta información se mueve hacia abajo a través de las capas de Enlace y Física del lado que envía, pasa por el cable de red, y sube por las capas de Física y Enlace del lado de el receptor hasta llegar a la capa de red de la computadora B. La interacción entre las diferentes capas adyacentes se llama interfase. La interfase define que servicios la capa inferior ofrece a su capa superior y cómo esos servicios son accedados. Además, cada capa en una computadora actúa como si estuviera comunicándose directamente con la misma capa de la otra computadora. La serie de las reglas que se usan para la comunicación entre las capas se llama protocolo.

Niveles en la pila TCP/IP.

Una red es una configuración de computadoras que intercambian información. Pueden proceder de una variedad de fabricantes y es probable que tenga diferencias tanto en hardware como en software, para posibilitar la comunicación entre estas es necesario un conjunto de reglas formales para su interacción. A estas reglas se les denominan protocolos. Un protocolo es un conjunto de reglas establecidas entre dos dispositivos para permitir la comunicación entre ambos. El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, mini computadoras y computadoras centrales sobre redes de área local y área extensa. TCP / IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET una red de área extensa del departamento de defensa.

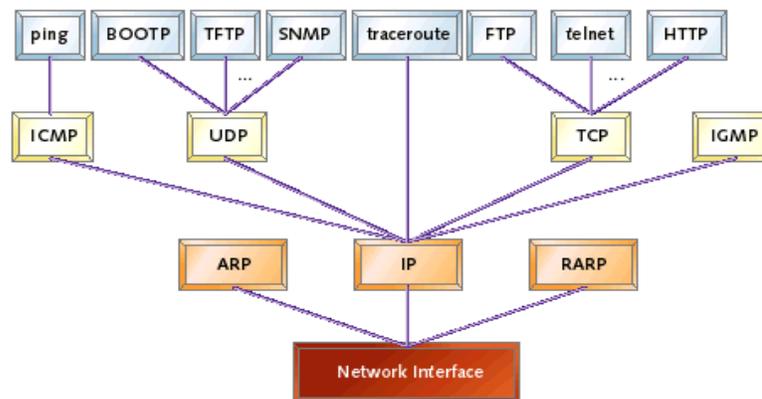


Figura 1.16: El protocolo TCP/IP.



Hay algunas discusiones sobre como encaja el modelo TCP/IP dentro del modelo OSI. Como TCP/IP y OSI no están delimitados con precisión no hay una respuesta que sea la correcta. El modelo OSI no es lo suficientemente rico en los niveles inferiores como para detallar la auténtica estratificación en niveles: necesitaría tener una capa extra (el nivel de Interred) entre los niveles de transporte y red. Protocolos específicos de un tipo concreto de red, que se sitúan por encima del marco de hardware básico, pertenecen al nivel de red, pero sin serlo. Ejemplos de estos protocolos son el **ARP** (Protocolo de resolución de direcciones) y el **STP** (Spanning Tree Protocol). De todas formas, estos son protocolos locales, y trabajan por debajo de las capas de Interred. Cierto es que situar ambos grupos (sin mencionar los protocolos que forman parte del nivel de Interred pero se sitúan por encima de los protocolos de Interred, como **ICMP**) todos en la misma capa puede producir confusión, pero el modelo OSI no llega a ese nivel de complejidad para ser más útil como modelo de referencia. El siguiente diagrama intenta mostrar la pila TCP/IP y otros protocolos relacionados con el modelo OSI original:

7	Aplicación	Ej. HTTP, SMTP, SNMP, FTP, Telnet, SSH y SCP, NFS, RTSP
6	Presentación	Ej. XDR, ASN.1, SMB, AFP
5	Sesión	Ej. TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS
4	Transporte	Ej. TCP, UDP, RTP, SCTP, SPX
3	Red	Ej. IP, ICMP, IGMP, X.25, CLNP, ARP, RARP, BGP, OSPF
2	Enlace de datos	Ej. Ethernet, Token Ring, PPP, HDLC, Frame Relay, ISDN
1	Físico	Ej. cable, radio, fibra óptica

1.10 El modelo de negocio para transmisión de archivos.

Como ya se mencionó, las instituciones financieras dependen en gran medida de la información que son capaces de procesar sin cometer errores, asimismo, esta información se debe distribuir en el momento oportuno a las personas y organismos adecuados.

Un ejemplo es la banca como tal, considérese el caso donde el departamento de crédito de un banco procesa diariamente (también existen otras periodicidades de proceso) la situación de la cartera, esta situación debe informarse a BANXICO (banco de México) para clasificar la misma en Vigente, Transitoria o vencida y determinar así, el riesgo de impagos, la clasificación debe ser enviada a BANXICO de manera oportuna y clara, es necesario contar entonces con un sistema como el STA.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

A partir de este momento, tómesese como ejemplo para este proyecto una AFORE (Administradora de Fondos para el Retiro), se puede hablar de que es importante y obligatorio mantener informada a la CONSAR (Comisión Nacional del Sistema de Ahorro para el Retiro) sobre todos los movimientos que existe por concepto de recaudación, retiros, traspasos, etc. Esta información obviamente es de gran volumen y es de proceso diario (También existen periodicidades de proceso semanal, mensual, bimestral, trimestral y anual entre otras) por lo que los organismos implicados solicitan y exigen los datos a través de transmisiones confiables y de gran eficacia para mantener una operación óptima.

Actualmente un negocio muy viable es la Administración de fondos para el retiro de los mexicanos, las afores son empresas financieras debidamente autorizadas por la S.H.C.P. y son supervisadas por la CONSAR que se especializan en el manejo de los ahorros para el retiro de los trabajadores. Una AFORE administra el ahorro para el retiro de todo aquel trabajador afiliado al IMSS que solicite sus servicios. Las Afores no son un banco, ni una aseguradora. Son instituciones que se dedican exclusivamente a administrar los recursos depositados en la cuenta individual de los trabajadores y tienen el objetivo de ofrecer al trabajador una pensión en el momento de su retiro.

Las situaciones anteriores ilustran de manera clara como un sistema de transmisión de archivos es sumamente necesario e importante para la operación de las instituciones financieras, se entiende que, son estos organismos quienes convienen el formato o Layout que deberán tener los registros que conforman los archivos que se intercambiarán. Lo mismo ocurre con los horarios de envío y recepción, aunque generalmente, son las autoridades bancarias las que determinan estos.

Actualmente la gran mayoría de las instituciones bancarias operan con un Layout y un horario de envío / recepción ya establecidos por las dependencias gubernamentales, lo cual es muy limitante para procesar y recopilar la información a transferir, en muchas ocasiones, los datos no están listos para cuando se necesitan ser enviados, lo cual lleva al personal de operaciones o usuario a dos opciones:

- Esperar al siguiente día cuando el siguiente proceso batch esta listo a enviar la información (se debe tomar en cuenta que el tiempo es valiosísimo).
- Solicitar al personal de sistemas se efectuó una transmisión manual y depender de la disponibilidad de estos (Muchas veces es más rápido esperar al otro día).

Obviamente ambas opciones son costosas, sería otra cosa totalmente diferente si el usuario en lugar de depender de un sistema automático (Batch) o de una persona de sistemas que efectuó la transmisión desde una compleja consola, pudiera hacerlo por él mismo a través de un amigable sistema de transmisión de archivos vía NDM.



CAPÍTULO 2

En el capítulo anterior se dejó entrever el problema que este trabajo resuelve, este capítulo se ocupará de abundar en el tema.

El objetivo principal del presente apartado es dar a conocer las necesidades de una AFORE en cuanto a transmisión de datos se refiere, de tal forma, que se pueda desarrollar un sistema que pueda solventar las mismas.

2.1 Delimitación del problema.

Problemática y Requerimientos

Como ya se mencionó, una Afore debe informar a la CONSAR sobre sus movimientos por concepto de recaudación, traspasos, retiros, etc. Actualmente los usuarios encargados de enviar esa información a dicha dependencia gubernamental lo hacen de la siguiente manera:

- Se generan los archivos desde la mañana y sin embargo, se envían por la noche automáticamente mediante un proceso Batch. Si el proceso por la noche no se ejecuta correctamente (cancela) se relanzará al día siguiente lo que implica pérdida de tiempo y en consecuencia una multa por parte de CONSAR.
- Cuando se trata de un envío de información en horario extraordinario el usuario puede acudir al personal de sistemas para que efectúe el envío ejecutando en ese momento el proceso Batch de transmisión, corriendo el riesgo de que sea atendido inoportunamente lo que de nuevo repercute en pérdida de tiempo y una probable multa por parte de CONSAR.

Necesidades

La AFORE requiere transmitir los siguientes archivos:

Nombre Lógico

AFILIACION SEMANAL
 ADMON DE CTAS INDIV PREST DE SERVICIOS
 TRASPASOS
 REGISTRO SAR 97
 ADMON. CTAS. INDIV. DE AFORE
 REPORTE DE PROMOTORES (MENSUAL)
 REPORTE DE PROMOTORES (QUINCENAL)
 BALANZA DE AFORE
 APORTACIONES VOLUNTARIAS
 BALANZA DE SIEFORE 2
 NETO AFORE-AFORE
 TRASPASOS LIQUIDACION

Nombre físico en HOST

DSIFNB.BAFPR.AFILIA.SEMANAL.BAF425A1
 DSIFNB.BAFPR.PRESTADO.A043
 DSIFNB.BAFPR.TRASPASO.A096
 DSIFNB.BAFPR.SAR97
 DSIFNB.BAFPR.INDIVIDU.A041
 DSIFNB.BAFPR.PROMOT.MENSUAL
 DSIFNB.BAFPR.PROMOT.QUINCENA
 DSIFNB.BAFPR.SIEFORE.A061
 DSIFNB.BAFPR.APORTA.CONSAR
 DSIFNB.BAFPR.SIEFORE.A124
 DSIFNB.BAFPR.NETO.AFORE.AFORE
 DSIFNB.BAFPR.TRASPASO.LIQUIDAC



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

TRASPASOS GESTION PRESTADORA
TRASPASOS GESTION CEDENTE
ENVIO DE PROMOTORES
TENENCIA Y OBTENCION DE PRECIO SIEFORE
TENENCIA Y OBTENCION DE PRECIO SIEFORE 2
BALANZA DE SIEFORE 1

DSIFNB.BAFPR.GESTION.PRESTADO
DSIFNB.BAFPR.GESTION.CEDENTE
DSIFNB.BAFPR.PROMOT.BAFK73A3
DSIFNB.BAFPR.SIEFORE.A071
DSIFNB.BAFPR.SIEFORE.A125
DSIFNB.BAFPR.SIEFORE.A062

Lo que el STA resuelve perfectamente, debido a que el usuario:

- Podrá enviar y recibir archivos a cualquier hora del día desde su propia Terminal usando una interfaz amigable.
- No dependerá del tiempo del personal de sistemas.
- No necesita el proceso nocturno y esperar al otro día a saber si concluyó correctamente, puede hacer él mismo la transmisión antes de que concluya el horario de trabajo y conocer en línea, si la transmisión fue exitosa.

Para lograr que el STA se desarrolle y se implemente, una AFORE cuenta ya con la infraestructura necesaria para soportarlo. Cuenta ya con el software indispensable para realizar la transmisión (NDM), un manejador de base de datos capaz de ser un repositorio común que mantenga la integridad, confidencialidad y seguridad de la información (DB2), un monitor de transacciones de gran poder para manejar gran volumen de concurrencias (CICS) y la plataforma que soporta lo anterior: IBM Mainframe.

De la misma forma, se hace necesaria una reclasificación de los archivos a enviar así como los archivos a recibir, de esta manera, se facilitara una pronta carga inicial en la base de datos a la que accederá para su operación el STA.

Finalmente, es importante redefinir las políticas internas dentro del negocio para que el personal que ahí labora en una primera instancia este enterada de las normas requeridas para manipular y alimentar el nuevo sistema y así, posteriormente, llevar a cabo las actividades regulares de transmisión de archivos mediante el STA. Tales políticas involucran aspectos de seguridad como usuarios con permiso de operación y reglas de acceso al lugar donde se encuentra la Terminal con el STA instalado.

Se pretende hacer un sistema que sea capaz de enviar y recibir archivos en línea que optimice la operación de la AFORE en general. Se desea que lo puedan emplear varias personas al mismo tiempo sobre una red local de terminales de Mainframe con una arquitectura centralizada. Debe ser un sistema que no necesite conocimientos especiales en computación para poderlo manejar. Es importante que implemente métodos para que el envío de los datos se realice de una manera ágil.

Se empleará el sistema operativo MVS que corre en un sistema Mainframe OS/390, DB2 y CICS, son todos productos propietarios por lo que, se contará con soporte técnico por parte de sus fabricantes.

El sistema debe mostrar mensajes al usuario que indiquen el status de la operación del mismo y debe emplear un módulo de seguridad para las distintas actividades dentro de la organización.



Objetivos específicos del proyecto

El STA debe ser un sistema amigable y facilitar su operación al usuario por lo que debe:

- Presentar en línea mediante un listado los archivos a enviar y los acuses a recuperar, de tal forma que el usuario simplemente seleccione la acción (envío o recepción) y el archivo.
- Agilizar la búsqueda y disponibilidad de los archivos a enviar.
- Reducir la incertidumbre del usuario en relación con una transmisión, es decir, el sistema debe ser capaz de avisar en línea si la transmisión fue exitosa o no.
- Reducir o eliminar el número de infracciones (multas) a las que se hace acreedora la AFORE cada vez que una transmisión con los procesos actuales no cumplen su objetivo.

Factores de éxito

- Comunicación y conocimiento de los objetivos, características y alcances del proyecto por parte de todos los involucrados.
- Contar con personal suficiente con conocimientos en las herramientas de análisis, diseño, desarrollo y pruebas a utilizar.
- Infraestructura robusta de cómputo y telecomunicaciones, así como en las herramientas y software de desarrollo.
- Eficiente administración y seguimiento de actividades.
- Retroalimentación oportuna de los usuarios.
- Identificación oportuna de riesgos y aplicación de acciones preventivas, y en su caso, correctivas.



Plan de actividades

FASE	DURACIÓN (HORAS)
DISEÑO	
Proceso Batch a invocar	
Especificación de componentes	32
Diagrama de secuencia Proceso	08
Diagrama de flujo Programa Batch	04
Diseño de tarjetas de transmisión	04
Lógica de Datos (BD)	
Diagrama entidad relación	16
Diccionario de datos	08
Objetos de la Base de datos	08
Aplicación en Línea	
Especificación de componentes	24
Interfaces de usuario (Mapas)	16
Diagrama de estados de programas Línea	08
Revisión detallado de diseño	08
DESARROLLO	
Codificación de proceso Batch (JCL)	08
Creación de interfaz de usuario	08
Codificación de programa Batch	08
Codificación de programas línea	24
Catalogación de tarjetas	08
Creación de la Base de datos	16
Carga inicial de datos	04
IMPLEMENTACIÓN	
Instalación en ambiente alterno	08
Prueba unitaria	16
Prueba Modular	24
Instalación en Producción	24
Documentación	40
Capacitación	40



2.2 Factibilidad

Factibilidad Técnica.

La tecnología de software en la actualidad se compone de dos grandes bloques: software libre y software propietario.

El software libre representa un nuevo paradigma en los sistemas de operación puesto que el desarrollo se hace por todas las personas en el mundo interesadas en incrementar sus capacidades, y lo hacen de una forma gratuita. Existe actualmente toda clase de software libre desde los sistemas operativos para grandes computadoras como Linux, hasta la más sencilla utilidad. El software libre por el lado del precio tiene muchísimas ventajas puesto que no cuesta, y es muy importante para el aprendizaje universitario, sin embargo, tiene algunas desventajas como por ejemplo, la relacionada al escaso soporte técnico existente.

Por otro lado el software que se denomina propietario, es el que adquiere de las grandes compañías como Microsoft, Oracle, IBM, HP, etc. A diferencia del software libre, el propietario tiene un costo por licencia de uso, sin embargo proporciona la asistencia técnica necesaria y existe documentación técnica de todo tipo sobre ellos en el mercado.

El STA utiliza el software propietario de IBM debido a la extensa documentación que existe en el mercado, esto es conveniente para resolver cualquier eventualidad que se presente a lo largo del desarrollo del mismo.

El sistema operativo MVS es el ejecutado en las grandes máquinas que utilizan las instituciones financieras y una AFORE, no es la excepción.

En cuanto al manejador de Base de Datos, al monitor de transacciones y a las herramientas de programación, se usan también las del mismo fabricante IBM debido a que son obviamente, las que mejor explotan los recursos del sistema operativo MVS.

La arquitectura se basa en un Host (Mainframe) y sus terminales y es la estructura organizativa del sistema, que incluye su descomposición en partes, conectividad, mecanismos de interacción y principios de guía que proporcionan información sobre el diseño del mismo.

Factibilidad operacional

Consiste en conocer si hay recursos humanos calificados y si el sistema instalado trabajara en perfectas condiciones.

Actualmente en la AFORE se cuenta con personal calificado para que opere el STA, debido a que, ya están familiarizados con el ambiente transaccional CICS.

Por otra parte se proporcionara asistencia técnica al usuario para resolver cualquier problema que pudiera presentarse en la operación del sistema.



2.3 Alcances del sistema

El STA tiene la siguiente funcionalidad y alcance técnico:

- El sistema se divide en dos bloques: Módulo Batch y Modulo Línea.
- La parte línea sin la parte Batch es inoperable y viceversa.
- Una vez instalados ambos módulos, el usuario podrá:
 - Enviar (Archivos de operación)
 - Recibir (Acuses de transmisión correcta)
- El modulo Batch es un JCL común y corriente, conformado de una serie de pasos en los cuales ejecuta un programa Batch para armar una tarjeta de transmisión que contiene origen y destino, así como el archivo a enviar.
- El modulo línea no es otra cosa que la interfaz de usuario, un menú y la pantalla que presenta los archivos y acuses para su transmisión. Es muy importante enfatizar que además de fungir como interfaz, invoca a la parte Batch para efectuar la transmisión. La manera de invocar se hace a través de una estructura de datos conocida como QUEUE donde se deja la información necesaria para que el sistema operativo interprete que se quiere lanzar un proceso Batch.
- El módulo línea implementa dos capas de seguridad, uno a nivel CICS a cargo de RACF y otro a nivel aplicativo a cargo del STA.

Para aumentar la eficiencia de una AFORE al intercambiar información con otros organismos financieros, es necesario contar con una herramienta que lo permita de una manera amigable, en línea y sin errores.

El STA propone la solución a la problemática planteada en este capítulo, se trata de un sistema que permita al usuario transmitir archivos durante y a cualquier hora de oficina.

El sistema cuenta con dos medidas de seguridad, la de RACF y la propia, a nivel aplicación.

La operación será en línea y permitirá al usuario independizarse del personal de sistemas, ya que, no será necesario requerirles que se efectúe una transmisión y tampoco tendrá que esperar al proceso nocturno, podrá hacerlo él.

Finalmente, se asegura que las multas por no entregar a tiempo la información requerida, se disminuirán drásticamente.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÒN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

EL USUARIO RECURRE AL PERSONAL DE SISTEMAS PARA UNA TRANSMISIÒN DE DATOS.

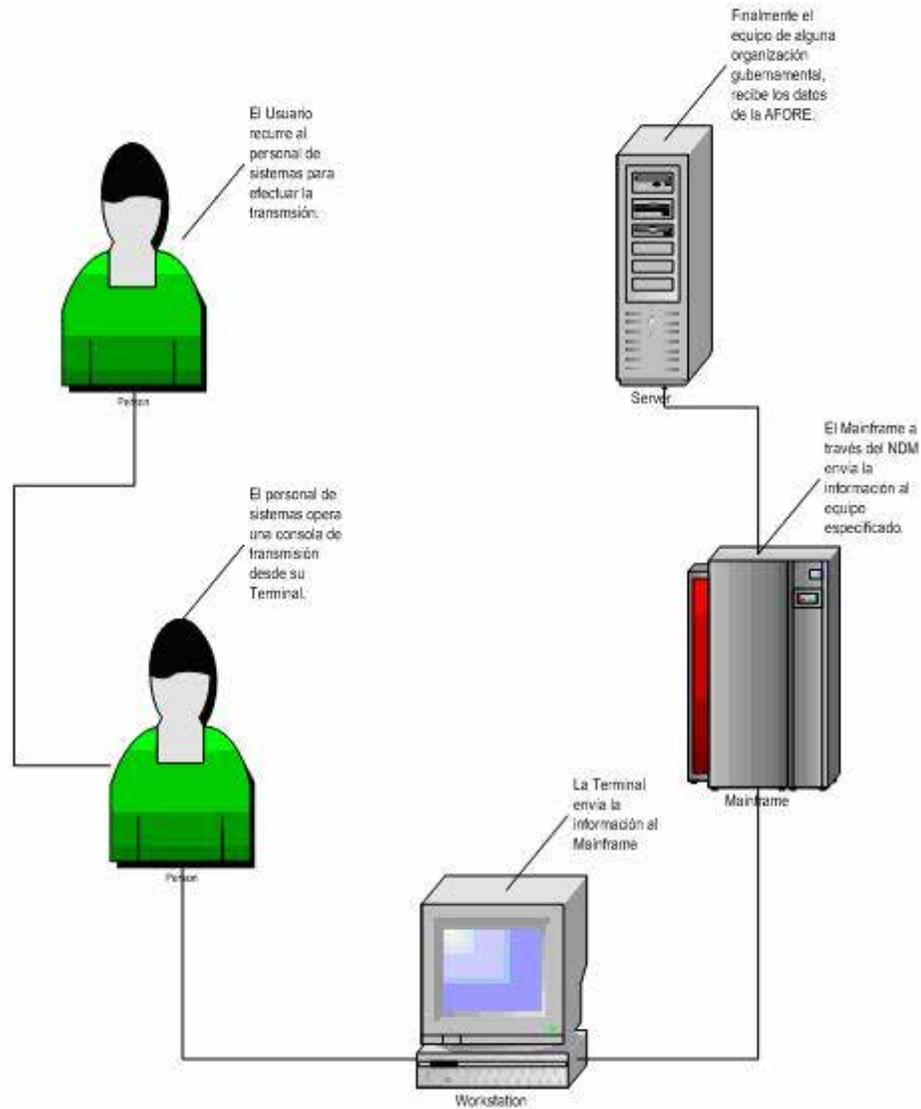


Figura 2.1: El usuario recurre al personal de sistemas cuando necesita realizar una Transmisión (envío o recepción de datos), esta es la dependencia que el STA eliminará.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÒN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

POR LA NOCHE, YA ESTÁN PROGRAMADOS LOS ENVÍOS Y RECEPCIONES DE INFORMACIÓN. NO INTERVIENE YA NI EL PERSONAL DE SISTEMAS NI EL USUARIO.

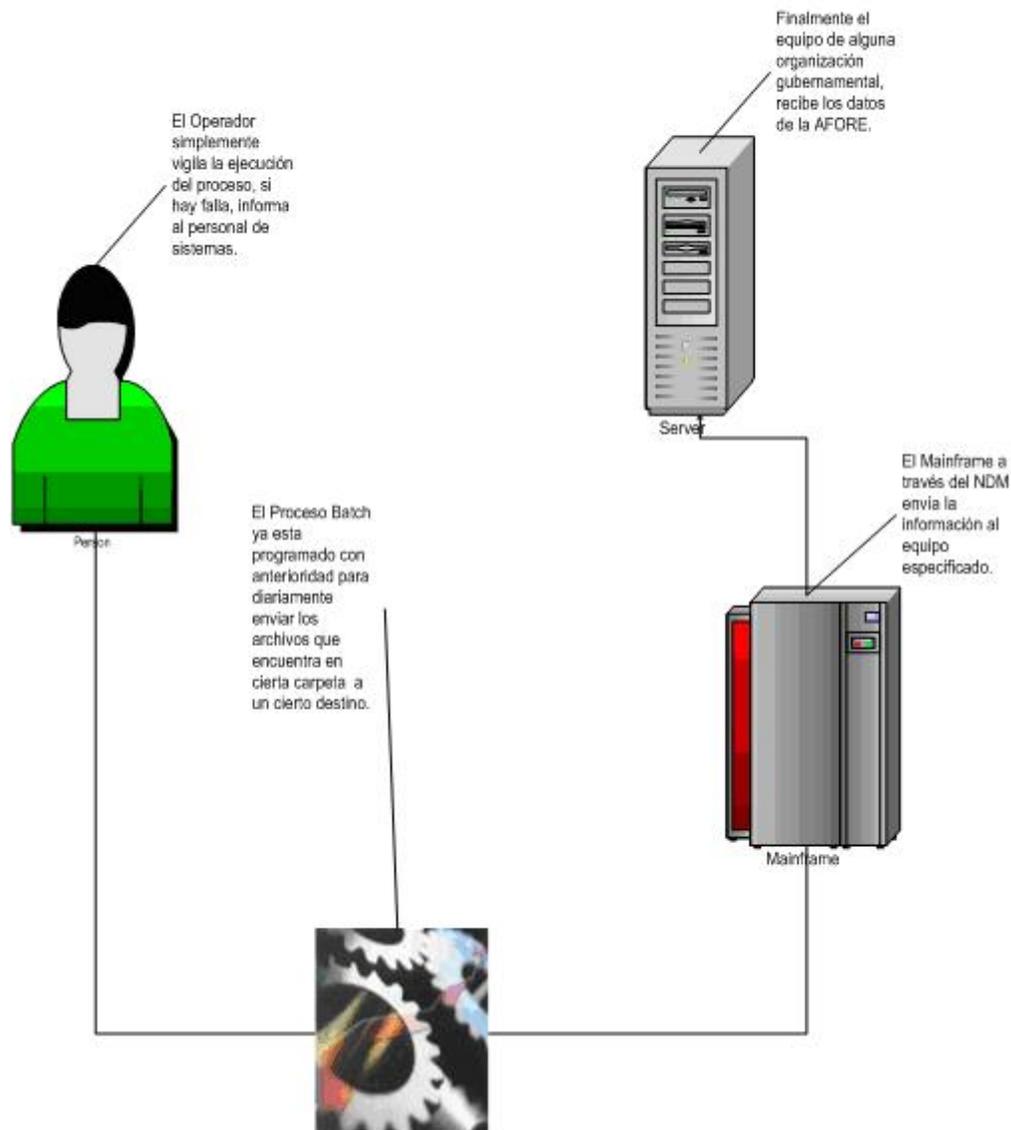


Figura 2.2: Ni el usuario ni el personal de sistemas interviene para realizar una Transmisión (envío o recepción de datos), lo hace automáticamente un proceso Batch, la desventaja es que ya esta programado a correr a determinada hora por la noche, si se requiere enviar un archivo por el día, el proceso no es la vía, el usuario tendría que recurrir al personal de sistemas como en la Figura 2.1 .



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÒN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

POR LA NOCHE, YA ESTÁN PROGRAMADOS LOS ENVÍOS Y RECEPCIONES DE INFORMACIÓN. NO INTERVIENE YA NI EL EL STA PERMITE AL USUARIO ENVIAR Y TRANSMITIR ARCHIVOS (EN TIEMPO REAL, EN LÍNEA) SIN DEPENDER DEL PROCESO NOCTURNO NI DEL PERSONAL DE SISTEMAS

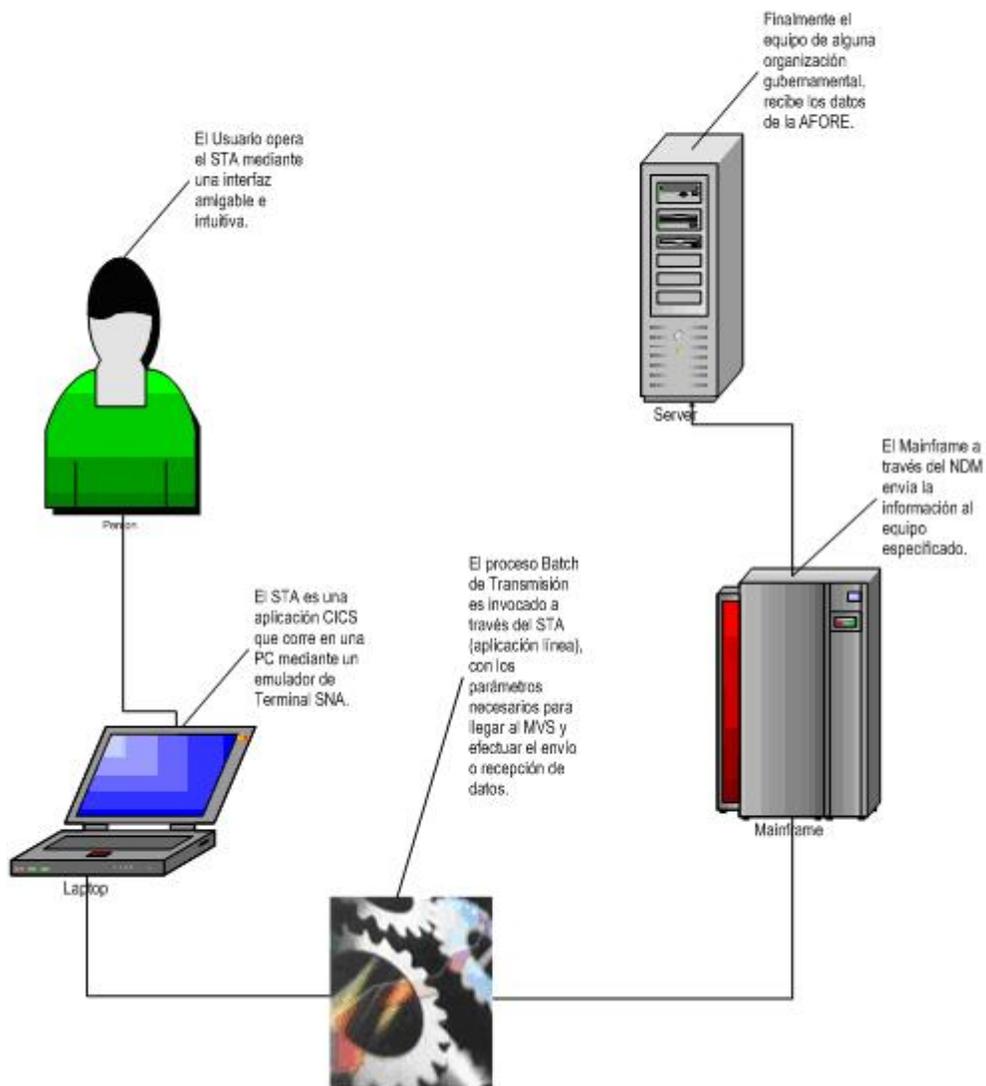


Figura 2.3: Gracias al STA, el usuario es capaz de Transmitir datos a la hora que lo desea, desde su propia PC conectada a la red, sin depender del proceso nocturno, ni del personal de sistemas, el resultado de la transmisión lo conoce de inmediato. Como se puede observar, el STA incrementa sobremanera la productividad del usuario, permite que el personal de sistemas se concentre en otras actividades y evita multas para la AFORE ante los organismos gubernamentales.



CAPÍTULO 3

El STA esta pensado en este proyecto para un Mainframe, su diseño conceptual no lo limita a la plataforma, y esto se demostrará en el siguiente capítulo, entre tanto, a continuación se muestra el diagrama del STA para una arquitectura HOST:

3.1 Diagrama conceptual del sistema¹.

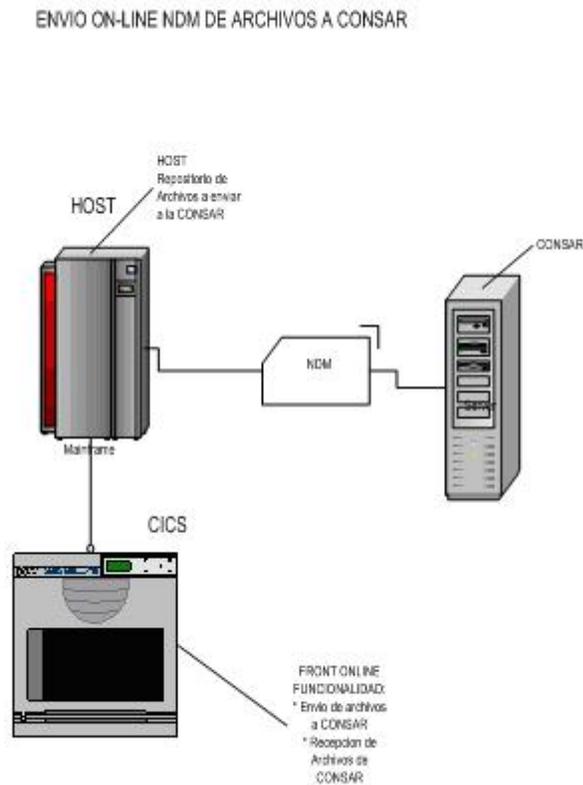


Figura 3.1: Diagrama Conceptual del STA.

En el apartado de RESUMEN, TABLAS Y FIGURAS a partir de la página 101 se muestran los casos de uso² y la arquitectura en tres capas del STA.

¹ El diagrama conceptual es la representación gráfica de los atributos, conceptos y asociaciones más importantes del dominio del sistema que se usará para resolver el problema.

² Descripciones narrativas en lenguaje natural de los procesos del dominio del sistema en un formato estructurado de prosa. Describen una secuencia de acciones.



3.2 Definiendo y estructurando el JCL de transmisión.

El JCL como ya se vio en el capítulo de antecedentes es un procedimiento que contiene varios pasos para lograr una tarea, el proceso que servirá para realizar la transmisión debe ser capaz de ejecutar los siguientes pasos:

- **PASO 00:** Verificar que el DB2 se encuentre activo para poder acceder a él mediante la utilería IKJEFT01, ésta es una herramienta propia del sistema que se emplea para esta tarea. En caso de que el DB2 no este activo, se terminará el proceso con el valor “08”, lo cual indica un error y en ese momento se terminará el Job informando en la consola lo ocurrido.
- **PASO 10:** Este paso se ejecuta sólo si el anterior regresó un código de retorno “00 (Éxito, DB2 activo), y es capaz de ejecutar un programa Batch que se encarga de acceder a la entidad que almacena el nombre de los archivos a transmitir para armar con esos datos la tarjeta de envío o recepción de archivos.
- **PASO 20:** Este paso se ejecuta sólo si el anterior regresó un código de retorno “00” (Éxito en el programa que arma la tarjeta de envío o recepción), y es capaz de hacer uso de la tecnología NDM para enviar o recibir un archivo mediante la tarjeta que se armo en el paso anterior.
- **PASO 30:** Este paso se ejecuta sólo si el anterior regresó un código de retorno “00” (Éxito en el envío o recepción del archivo), y es capaz de ejecutar un programa Batch que sirve como retardador, con el fin de que pasen un par de minutos después de la transmisión para poder recuperar la consola del sistema que informa el status en que termino el envío o recepción (no es lo mismo que el paso de transmisión termine bien a que no hayan presentado errores en la transmisión, para eso se recupera la consola).
- **PASO 40:** Este paso se ejecuta sólo si el anterior regresó un código de retorno “00” (Tiempo transcurrido con éxito en el programa retardador), y es capaz de recuperar la consola que informa el resultado de la transmisor y alojarla en un archivo de HOST.
- **PASO 50:** Este paso se ejecuta sólo si el anterior regresó un código de retorno “00” (Creación del archivo que aloja la consola de transmisión), y es capaz de ejecutar otro proceso NDM que ahora toma el archivo que contiene la consola y la envía mediante una tarjeta de parámetros NDM al servidor NT donde el usuario puede consultar el status de la transmisión.



3.3 Definiendo los parámetros que el sistema recibirá

Tómese en cuenta que el STA esta conformado por dos partes, la parte línea y la parte Batch, el proceso del punto anterior obviamente es parte del módulo Batch debido a que es un procedimiento automatizado que es invocado por la parte línea como se verá más adelante.

El uso de la tecnología NDM se realiza mediante un paso del anterior procedimiento, se sustenta en la sintaxis que se presentará más adelante, pero en este momento es muy importante mencionar que dentro de esa sintaxis se hace referencia a una tarjeta de NDM que no es otra cosa que un archivo de parámetros. Esta tarjeta es armada con el programa Batch que se ejecuta también en el procedimiento anterior, el programa accede a la base de datos para obtener los parámetros necesarios para armar dicha tarjeta, y la base de datos se alimenta precisamente, de lo que el usuario captura en el front de transmisión en línea. Finalmente se puede concluir, los parámetros que el proceso Batch recibe, son los que el usuario capturó en el módulo línea a través de una pantalla. Los parámetros son los siguientes:

- Acción: Envío o recepción.
- Nombre de archivo: Nombre físico en Host mas fecha del día.

De tal manera que después de que el usuario opera la pantalla línea para elegir la acción (recibir o enviar) y el archivo o acuse y pulsa la tecla <Enter>, en ese momento, se arma una QUEUE con los datos necesarios (nombre de JCL) para que el sistema operativo interprete que se ejecutará un proceso, además, a través de la pantalla la parte línea escribió en la entidad de base de datos los parámetros necesarios para crear la tarjeta de transmisión NDM que se empleara en el paso 20 del proceso Batch para efectuar el envío o recepción del archivo. La tarjeta que se arma tiene la siguiente forma:

```
COPIA      PROCESS SNODE=INSCONS.UNIX
STEP1     COPY   FROM(DSN=TBCODCM.A020827.A001 -
                DISP=SHR) -
                TO(DSN="/export/home/rec/AFORES/A554/RECEPCION/20020827.001" -
                disp=rpl snode -
                sysopts=":strip.blanks=no:datatype=text:xlate=yes:")
```

Donde:

COPIA:	Nombre del proceso.
PROCESS NODE:	Nodo del destino.
STEP:	Número de paso (puede haber N pasos).
COPY FROM:	Ruta de HOST origen.
TO:	Ruta de servidor destino.
DISP:	Atributo RPL: reemplaza archivo si ya existe en destino.



3.4 Diseño de la pantalla (Front de transmisión).

Como se mencionó en el capítulo de antecedentes, el CICS proporciona utilerías y soporte para el diseño de la interfaz (pantalla) con el usuario.

Esta interfaz de usuario es modo texto y es operable mediante las teclas de navegación y de función.

Se definen dos pantallas, una servirá como Menú (para agregar escalabilidad al sistema, en el futuro podría haber más de una opción, entre tanto, el menú sólo tiene una opción) y otra que servirá para presentar los archivos disponibles para enviarse y los acuses que se pueden recuperar, es decir, la segunda pantalla, es el Front de transmisión que invocara al módulo Batch para efectuar la misma, todo ello, en línea.

El diseño del Front de transmisión se basa en 3 pantallas, la primera pantalla corresponde agregar la opción TRANSMISIONES al menú principal de la AFORE.

La siguiente pantalla ilustra el punto anterior.

```
1234567890123456789012345678901234567890123456789012345678901234567890
1
2                                AFORE
3                                M E N U   P R I N C I P A L
4
5
6 AC Aclaraciones
7 CA Cajeros
8 CO Contabilidad
9 EC Estados de Cuenta
0 MU Maestro de Usuarios
1 RC Recepción de Cuotas
2 RE Retiros
3 TR Traspasos
4 UN Unificación de Cuentas
5 SC Separación de Cuentas
6 PL Menú PLD
7 TM Transmisiones
8
9
0 FUNCION CLAVE
1 ==>
2 F03=M ANTER
3                                F12=TRAN ANTER   F13=SALIR   F16=M PRINCIPAL
4
```

Figura 3.2: Bosquejo del Menú principal del negocio.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"**

La segunda pantalla muestra el submenú de TRANSMISIONES pensado para que en el futuro pueda tener más opciones.

```

1234567890123456789012345678901234567890123456789012345678901234567890
1
2
3
4
5
6 01 Transmisión de archivos CONSAR
7
8
9
0
1
2
3
4
5
6
7
8
9
0 FUNCION CLAVE
1 ==> TM 01
2 F03=M ANTER
3
4
5 F12=TRAN ANTER F13=SALIR F16=M PRINCIPAL

```

Figura 3.3: Bosquejo del Menú de Transmisiones.

La tercera pantalla muestra el Front de TRANSMISIONES, desde aquí, el usuario puede enviar y recibir archivos.

```

1234567890123456789012345678901234567890123456789012345678901234567890
1
2
3
4 Acción...ç ! " " = Consulta E = Enviar R = Recibir
5 Archivo: ç ! Fecha (Nombre archivo) ç / / !
6 ç X ! Archivo1 Descripción X
7 ç ! .
8 ç ! .
9 ç ! .
0 ç ! .
1 ç ! .
2 ç ! .
3 ç ! .
4 ç ! .
5 ç ! .
6 ç ! .
7 ç ! .
8 ç ! .
9 ç ! ArchivoN Descripción X
0 Función Clave
1 ==> TM
2 F03=M ANTER F06=CANCELA F07=PAG ATRAS F08=PAG ADEL
3 F12=TRAN ANTER F13=SALIR F16=M PRINCIPAL
4

```

Figura 3.4: Bosquejo de la pantalla de transmisión.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Una vez codificadas, las pantallas muestran el siguiente aspecto:

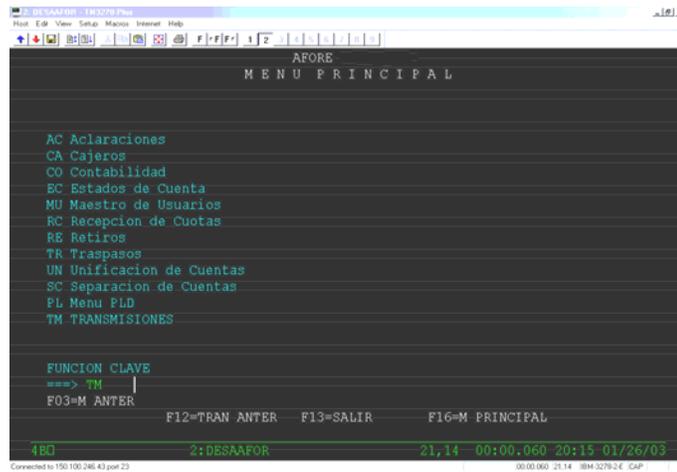


Figura 3.5: Clásico Menú principal de una AFORE. Se accede al STA por la opción TM.

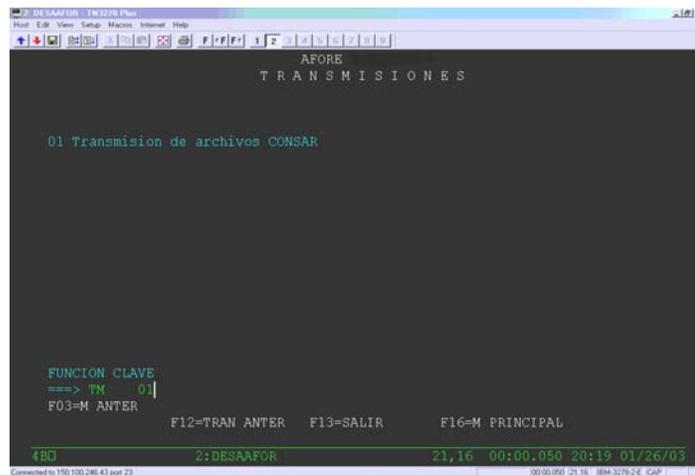


Figura 3.6: Menú del STA.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

- El nombre de la pantalla es BAFL56 y la funcionalidad la aporta el programa BAFPL56.
- Pantalla 3 Transmisión de archivos CONSAR:
 - Esta pantalla corresponde al Front de transmisión y es la que finalmente, una vez capturados todos los datos por parte del usuario, invoca a la parte Batch para efectuar la transmisión.
 - En esta pantalla, como se ha mencionado, el usuario ingresa los datos correspondientes para realizar la transmisión:
 - Acción: “Espacio” (consulta de archivos disponibles), “E” (Envío), “R” (Recepción).
 - Fecha (Nombre de archivo): Corresponde a la fecha del día, el nombre del archivo a enviar se compone del nombre físico del mismo más la fecha en el siguiente formato AAMMDD.
 - Archivo: Permite buscar un archivo en específico entre todos los disponibles, si este campo no se informa, la consulta enviará como resultado todos los archivos.
 - La consulta es de tipo paginación arriba abajo.
 - Más adelante se explicará la funcionalidad utilizando casos de uso.
 - Después de ingresar los datos anteriores y oprimir <Enter>, la pantalla guarda los parámetros en la Base de datos en la tabla TAF239 (que se describirá más adelante) y escribe el nombre del procedimiento a ejecutar en una QUEUE misma que el sistema operativo lee, interpreta y ejecuta, se debe recordar que es así como se logra invocar a la parte Batch, en ese momento comienza la ejecución automática del proceso, que en uno de sus pasos ejecuta el programa Batch identificado como BAFPM72, quien es el encargado de armar a partir de la tabla TAF239 (alimentada por el programa línea BAFPL57) la tarjeta NDM de transmisión, el procedimiento finalmente, ejecuta el paso encargado de la transmisión a partir de la tarjeta NDM.
 - El nombre de la pantalla es BAFL57 y el programa que le proporciona la funcionalidad es el BAFPL57.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

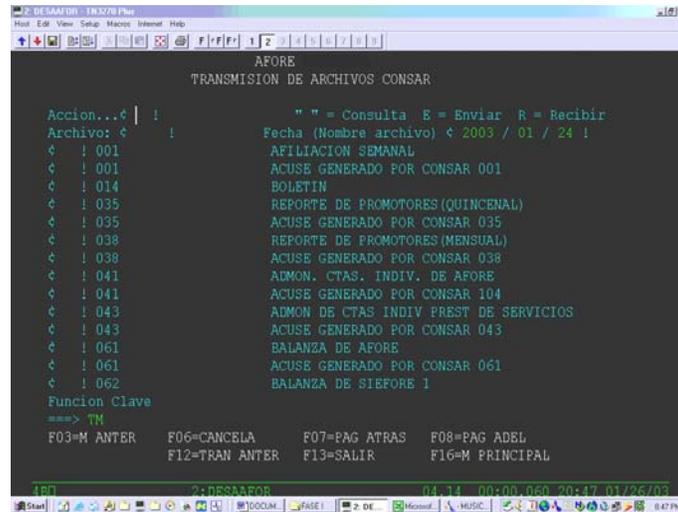


Figura 3.8: Aspecto del Front de transmisión, justo antes de ejecutar un envío.

3.5 Diagrama de estados de la aplicación línea (BAFPL57)

A continuación se describe la funcionalidad del programa encargado de desplegar los archivos disponibles a transmitirse y de invocar al módulo Batch. Analizando el siguiente diagrama, se podrá comprender mejor el comportamiento del programa línea.

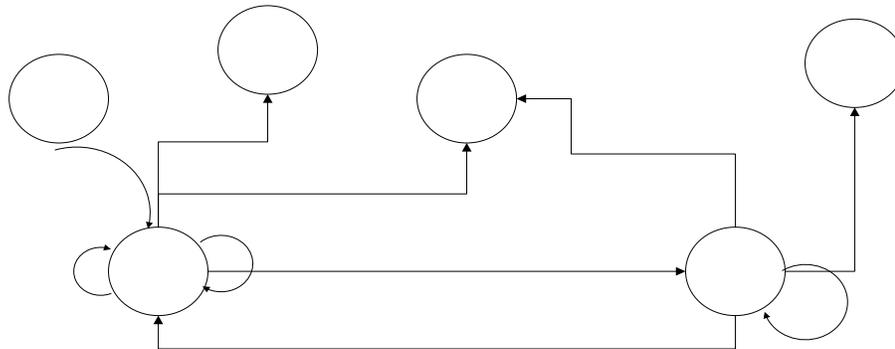


Figura 3.9: Diagrama de estados del programa Línea.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Estado 0.
Inicio de Conversación.

Estado 1.
Pantalla inicial o con datos erróneos.

Teclas de función válidas:

Enter – Ejecuta
F2 – Transacción Anterior
F3 – Menú Ant.
F5 – Cancela
F13 – Salir del STA

Estado 2.
Pantalla con datos de una consulta exitosa de archivos a transmitir, avance de página, regreso de página.

Teclas de función válidas:

Enter – Ejecuta
F2 – Transacción Anterior
F3 – Menú Ant.
F5 – Cancela
F7 – Página anterior
F8 – Página siguiente
F13 – Salir del STA

Estado 22.
Transferencia de control.

Estado 99.
Fin de la transacción.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

TRANSICIÓN	SERVICIO
t01 – Inicia conversación	Pantalla Inicial
t02 – F2 transacción anterior	Transfiere Control
t03 – F3 menú anterior	Transfiere Control
t04- Enter, transacción siguiente	Transfiere Control
t05- F13 salida del sistema	Transfiere Control
t06 – Enter, consulta exitosa	Pantalla consulta
t07 – Enter, error en consulta por llave invalida	Misma Pantalla (despliega mensaje de error)
t08 – F5, cancela	Pantalla Inicial
t09 – F7, regreso de página exitosa	Pantalla consulta
t10 – F8, avance de página exitoso	Pantalla Consulta
t11 – Transferencia de control fallida	Pantalla inicial

Figura 3.10: Interpretación del diagrama de estados de la aplicación en Línea de la figura 3.9.

3.6 Seguridad del Front de transmisión.

Como se mencionó anteriormente el módulo de seguridad del STA se implementa en 2 capas:

- RACF: Es la seguridad nativa que incorpora el sistema operativo al firmarse al monitor de transacciones CICS. El usuario que opera una pantalla dentro del CICS, previamente validó sus credenciales en el sistema (nombre de usuario y contraseña) mediante el sistema de seguridad RACF que verifica si el usuario puede ingresar al CICS.
- Aplicativa: El STA comprueba que el usuario que esta operando la pantalla de transmisión de archivos este autorizado para ello, esto se logra mediante la tabla de base de datos de parámetros (se describirá más adelante) donde se almacenan los usuarios autorizados.



3.7 Diagrama de flujo de la aplicación Batch.

El siguiente diagrama de flujo muestra el comportamiento del programa Batch que se encarga de armar la tarjeta de transmisión NDM.

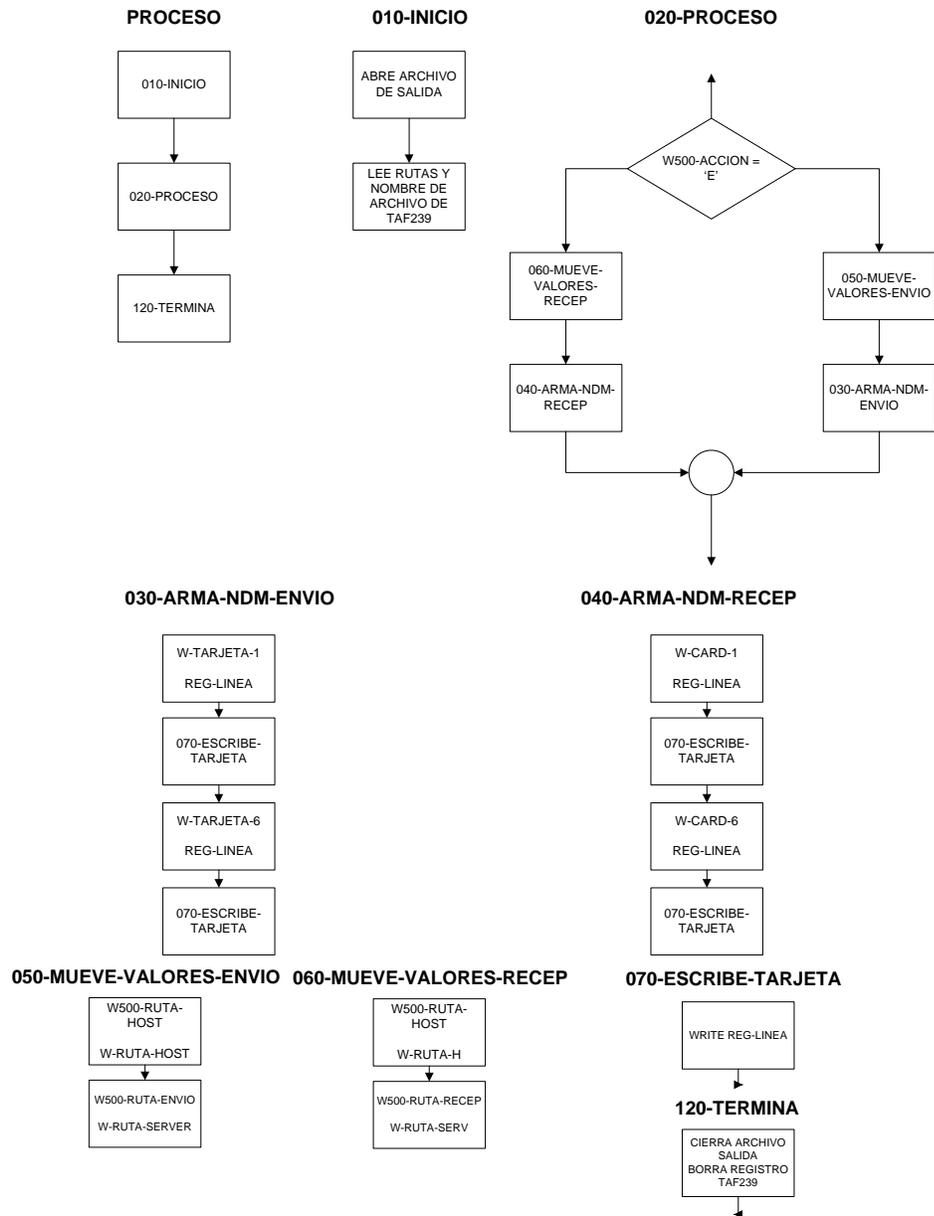


Figura 3.11: Diagrama de flujo aplicación Batch con COBOL.



3.8 Diseño de la entidad que almacena acciones y rutas de archivos

A lo largo del proyecto se ha podido observar que se hace referencia a la base de datos para alimentar y obtener los parámetros que servirán para realizar la transmisión.

Se creo una tabla de base de datos que contiene los parámetros que el usuario captura en el Front de transmisión NDM y que el proceso Batch lee para realizar la transmisión. Esta entidad se llama TAF239 y como se ha mencionado es una tabla de DB2.

Al crear la tabla se cuidaron los aspectos fundamentales del diseño de una entidad de base de datos relacional como son:

Redundancia: La redundancia de datos se refiere a la existencia de información repetida o duplicada en diferentes tablas de una base de datos.

Al ser el STA un sistema nuevo dentro de la organización, se determinó usar una tabla que permitiera la comunicación entre el módulo línea y el módulo Batch (pasar parámetros), los datos que se almacenan en esta tabla no existen en ninguna otra, así se evita caer en redundancia de datos.

La redundancia conduce a muchos problemas que tienen que ver con la integridad y consistencia de los datos. La redundancia de los datos requiere múltiples procedimientos de entrada y de actualización de los mismos.

Dentro de una base de datos relacional la redundancia debe ser mínima y controlada. En ocasiones existirán motivos válidos de negocios o técnicos para mantener varias copias de los mismos datos almacenados. En este proyecto, no es el caso.

Consistencia: Frecuentemente los problemas de consistencia de los datos se deben a la redundancia de los mismos.

Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar; ya que al modificar, eliminar o agregar un dato, en esas condiciones, debe realizarse en cada una de las instancias del mismo con el riesgo de no ser realizado en su totalidad, generando en este caso datos inconsistentes. Debido a que en este proyecto no se genera redundancia, difícilmente existirá inconsistencia.

Integridad: La integridad no sólo contempla el hecho de que los datos sean consistentes dentro de la base, sino además, que los valores que posean los datos sean válidos de acuerdo a las dependencias funcionales entre las tablas y de acuerdo a las políticas del negocio.

La inconsistencia de dos entradas que representan el mismo “hecho” es una falta de integridad que, por supuesto sólo ocurre si existe redundancia de datos almacenados, por lo tanto, en este proyecto la integridad de los datos esta asegurada y se logra mediante:



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

- Manteniendo una redundancia mínima y controlada.
- Estableciendo llave primaria en la nueva entidad (TAF239).
- Estableciendo reglas de validación durante la creación y edición de los datos.
- Estableciendo procedimientos que validen la dependencia funcional entre tablas relacionadas.

Reglas de Normalización: Estas reglas sirven para mantener la integridad referencial de la Base de datos, se describen a continuación cada una de ellas:

- Primera forma Normal. Se debe tener un atributo único que identifique a cada renglón de la tabla (En el caso de la TAF239 es el campo IDARCH). Se asigna una llave primaria a este atributo. Si se tiene una tabla T1 con atributos “a1”, “a2”, “a3”..., se dice que esta en primera forma normal si “a1” como clave identifica de manera única a cada renglón de T1.
- Segunda forma Normal. No deben existir dependencias parciales en una tabla, es decir, si una tabla tiene una llave primaria formada por uno o dos atributos, los atributos no llave deberán depender de todos los atributos que forman la llave.
- Tercera forma Normal. No deben existir dependencias transitivas, es decir, un atributo no llave, no debe depender de otro atributo no llave.

Reglas de Codd: Para crear una base de datos relacional es necesario cumplir con las reglas de Codd:

- Cualquier DBMS que proclame ser relacional, deberá manejar completamente las bases de datos por medio de sus capacidades relacionales.
- Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores de una tabla.
- Se garantiza que todos y cada uno de los datos en una base de datos relacional pueden ser leídos recurriendo a una combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.
- En un DBMS totalmente relacional se soportan los valores nulos para representar información faltante o no aplicable a una forma consistente independientemente del tipo de dato.
- La descripción de la base de datos se representa en un nivel lógico de la misma forma que los datos ordinarios, de tal suerte que los usuarios autorizados pueden aplicar el mismo lenguaje relacional para consultarla que aquel que emplean para con sus datos habituales.
- Dato completo. Contempla definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorización inicio y fin de una transacción.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

- Todas las vistas que teóricamente sean actualizables, deben ser actualizadas por medio del sistema.
- La posibilidad de manejar una relación base o una relación derivada como un solo operador se aplica a la lectura, inserción, modificación, y eliminación de los datos.
- Los programas de aplicación y la actividad de las terminales no deberán ser afectadas por cambios en el almacenamiento físico de los datos o en el método de acceso.
- Los programas de aplicación y la actividad de las terminales no deberán ser afectadas por cualquier situación que preserve la información y que teóricamente permita la afectación en la tabla base.
- Las restricciones de integridad de una base de datos deberán poder definirse en el mismo sublenguaje de datos relacional y deberán almacenarse en el catalogo, no en los programas de aplicación.
- Un DBMS relacional tiene independencia de la distribución.
- Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), ese nivel no puede ser utilizado para suprimir las reglas de integridad y las estrictiones expresadas en el lenguaje relacional de nivel superior (múltiples registro a la vez).

Para el análisis de la Base de datos es necesario seguir los siguientes pasos para obtener el modelo entidad-relación:

Pasos para modelar la Base de Datos.

1. Identificar las posibles entidades.
2. revisar que las entidades estén en el contexto del sistema.
3. Identificar los atributos de las entidades.
4. Revisar que los atributos estén en el contexto del sistema.
5. Comenzar a definir las llaves primarias.
6. Establecer las relaciones que existen entre las entidades (Asociación y tipos de relación).
7. Revisar que las relaciones estén en el contexto del sistema.
8. Aplicar primera y segunda forma normal. Romper asociaciones de muchos a muchos.
9. Documentación. Establecer reglas, defaults, constraints, índices, definir tipos de datos, revisar los rangos, revisar si los campos son o no requeridos.
10. Realizar pruebas de escritorio.

Los pasos anteriores son sólo una vía para llegar al modelo final, y no necesariamente se tienen que seguir en un orden secuencial, aunque si es recomendable.

El proceso se repite hasta que se logre el resultado esperado: Un modelo que represente la realidad de lo que se esta analizando.



Diccionario de Datos.

El contenido del diccionario de datos se considera “datos” acerca de los datos, y se conoce como Meta datos, es decir, definiciones de objetos de la base de datos, y no tan solo “Datos en bruto”. El diccionario de datos del STA puede apreciarse en el apartado del resumen, tablas y figuras.

Modelo Entidad-Relación

Modelo Propuesto

TABLA
BASE DE DATOS
APLICACIÓN

TAF239
PRODUCCION
AFORE

LLAVE PRIMARIA

		TAF239	IAF239P1
		IDARCHIVO	IDARCHIVO
		DESCARCH	ACCION
		ACCION	STATUSARCH
		RUTAHOST	FHACTIONAL
		SERVENVIO	
		SERVRECEP	
		NOMARCHIVO	
DETALLE		STATUSARCH	
LLAVE		FHACTIONAL	
		HMACTIONAL	
		CDUSUARIO	

CREATE TABLE

Figura 3.12: Definición de la tabla que contendrá las rutas origen y destino, así como, la acción: Envío, Recepción, Alta.



TABLA DE RUTAS Y
ACCIONES

TAF239	
PK	<u>IDARCHIVO</u>
PK	<u>ACCION</u>
PK	<u>STATUSARCHIVO</u>
PK	<u>FHACTUAL</u>
	DESCARCHIVO RUTAHOST SERVENVIO SEVRECEP NOMARCHIVO HMACTUAL CDUSUARIO

Figura 3.13: Modelo entidad relación de la tabla que contendrá las rutas origen y destino, así como, la acción: Envío, Recepción, Alta.

La carga de trabajo que tendrá el sistema se divide en dos: La primera la soporta la parte línea y la segunda, la parte Batch.

La parte línea como ya se mencionó, sirve como interfaz de usuario y además provee a la parte Batch de los datos necesarios para su funcionamiento.

La parte Batch recibe los parámetros de la línea y se encarga de realizar, como si fuera un proceso común y corriente, la transmisión NDM.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

CAPÍTULO 4



CAPÍTULO 4

Todo sistema debe regirse por el ciclo del software: análisis, desarrollo, pruebas, liberación y mantenimiento. En este capítulo se aprecia la manera en que el STA es sometido a la matriz de pruebas diseñada para cubrir las necesidades y expectativas del negocio y/o usuario.

La prueba de los programas es la parte del proceso de evaluación que suele realizarse durante la programación y también, en una forma distinta, cuando es terminado. La prueba consiste en ejecutar los programas simulando ser el usuario final para determinar si el sistema es amigable y de sencillo manejo y si es homogéneo en toda su presentación, entre otros aspectos.

Utilizando datos similares que habrán de ser suministrados al sistema, se observan los resultados y se deduce la existencia de errores o insuficiencias del programa a partir de las anomalías de estos resultados.

En ocasiones se piensa en la prueba y en la ejecución de programas como una misma cosa. Aunque están muy relacionados, en realidad son procesos distintos. La prueba es el proceso de establecer la existencia de errores en los programas; la depuración es el proceso de localizar donde se produjeron esos errores y corregir el código, optimizándolo.

Es muy importante comprender que la prueba nunca será capaz de demostrar que un programa es incorrecto. Siempre es posible que existan errores aún después de la prueba más completa. La prueba de programas sólo puede demostrar la presencia de errores, no su ausencia en el mismo.

4.1 Prueba Unitaria.

El objetivo de un plan de prueba unitaria es detectar un comportamiento distinto al que necesita el usuario y de lo que se planeó en el análisis, diseño y desarrollo, este tipo de prueba se aplica antes de integrar el nuevo desarrollo al resto del sistema.

Las pruebas que se aplican al sistema se basan principalmente en los documentos de análisis y diseño del STA, así como el manual de usuario y técnico en donde se comprueba que es una exacta referencia de lo que se necesita para el manejo del sistema.

Pruebas especiales del sistema.

Prueba de carga máxima: Se basa en la existencia de tiempos críticos en los sistemas en línea, es decir, la respuesta de un sistema en prueba cuando varios usuarios quieren acceder a él (conurrencia). En el caso del SAT esta prueba muestra resultados satisfactorios debido a la facilidad de manejo de transacciones en línea que ofrece el CICS, como se mencionó en el capítulo 1.

Pruebas de almacenamiento: Mediante esta prueba se determina si el sistema realmente soporta la capacidad considerada en su diseño de la misma que debe ser considerada antes de la implementación.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

Para ello se van almacenando datos en forma continua hasta alcanzar la capacidad máxima teórica. La capacidad real se obtiene haciendo la comprobación respectiva.

Prueba de recuperación: Esta prueba consiste en crear un evento de fallas o pérdida de datos, para que los usuarios vuelvan a cargar y operar a partir de una copia de respaldo.

Prueba de procedimientos: Con esta prueba se determina si los manuales de documentación y ejecución contiene una descripción detallada y si refleja realmente las acciones que se llevan a cabo para el funcionamiento del sistema. Para ello el usuario debe seguir las instrucciones exactas como lo indica el manual; en la sección de Resumen, tablas y figuras de este trabajo se puede apreciar el manual de usuario del STA.

Puntos de revisión

Acceso al sistema: Por el tipo de datos que maneja el sistema, es importante conocer la seguridad por cada persona que lo utiliza. En esta primera versión del STA, no existen los niveles de seguridad, todo se resume a usuarios con autorización para operarlo o usuarios sin autorización.

Para llevar a cabo esta prueba simplemente se realizó lo siguiente:

Se debe recordar que la seguridad para utilizar el STA se pasa por dos capas la de RACF (propia del CICS) y la aplicativa (por medio de una validación en una tabla de base de datos que contiene usuarios autorizados para el STA).

- Se firma al sistema un usuario sin autoridad para el CICS
 - o Obviamente al intentar autenticarse en el CICS para llegar a la pantalla de transmisión, recibe una negativa de acceso.
- Se firma al sistema un usuario con autoridad para el CICS
 - o El acceso al CICS es exitoso (permisos avalados por RACF), sin embargo, este usuario no se encuentra en la tabla PARAM_AF (tabla de parámetros del sistema donde se incluyen los usuarios autorizados) por lo que es rechazado por la capa de seguridad aplicativa del STA.
- Se firma al sistema un usuario con autoridad para el CICS
 - o El usuario accesa exitosamente al CICS y además puede operar sin contratiempos el STA debido a que existe en la tabla de parámetros PARAM_AF.

Estructura: En esta parte se evalúa la organización, distribución y el orden que guarda todo el sistema. Los puntos de revisión son:

- La uniformidad del diseño y nomenclatura que permita distinguir las pantallas de acceso al STA como son el menú, el submenú y el Front de transmisión.
- Estética y homologación de títulos, tipos de letra, colores, búsquedas y presentación de las pantallas.
- Entradas y salidas de cada pantalla que presente la información que se requiere
- Los mensajes de ayuda y de error correspondan a cada caso.



Operación del sistema.

En este apartado se mide que tan amigable es el sistema para el usuario y se verifica la correcta operación de los procesos.

La primera actividad que se realiza consiste en la secuencia y la navegación entre los registros y las funciones que realiza cada pantalla, éstos son:

- La introducción y selección de texto en cada campo correspondiente.
- Secuencia de llenado en orden por medio de tabuladores, <Enter> o alguna tecla de función, el suministro de información en una pantalla no puede ser aleatorio, debe seguir una secuencia lógica.
- Correcta secuencia de navegación entre pantallas, así como la navegación propia de la pantalla denominada Front de transmisión que consiste en el despliegue amigable de los datos que el usuario final podrá transmitir.
- La operación que se selecciona es la que se ejecuta (recibir o enviar un archivo).
- Navegación entre los registros desplegados en el Front de transmisión por medio de teclas de función como, ir a la primera página, a la anterior, a la siguiente y a la última.
- Verificación de las búsquedas de información para que correspondan con lo que se ha solicitado.

La segunda actividad es la navegación entre los menús:

- Fácil de usar y amigable.
- Verificación de entradas y salidas (enlaces entre menús y pantallas), que funcionen adecuadamente.
- Opciones de navegación en cada menú (entrar, salir y regresar).
- Revisión de las salidas que corresponden a cada parte del sistema.

Validación de datos y funcionalidad.

El objetivo de esta prueba es asegurar que los filtros o validación que se requiera para cada dato que se introdujo o la relación entre datos fueran válidos.

En esta sección se realizan las siguientes acciones:

- Revisar que no se acepten datos erróneos (se respete la definición de cada campo).
- Identificación de las fallas posibles al introducir errores intencionales.
- Identificación de los campos clave y verificación de que no se dupliquen.
- Identificación de los campos que pueden quedar sin datos.
- Integridad del sistema después de realizar una operación.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
 FACULTAD DE ESTUDIOS SUPERIORES ARAGON
 “SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
 PLATAFORMA MAINFRAME”**

4.3 Prueba Modular.

La prueba modular permite verificar la correcta comunicación que componen el sistema, que las salidas de un módulo correspondan a las entradas de otro según la finalidad del mismo.

En el STA la prueba unitaria es la que permite determinar si es posible la pronta y correcta implementación del mismo, ya que, por ser un sistema aislado e independiente del resto de la aplicación AFORE (debido a que tiene sus propias fuentes de suministro de información), no se necesita hacer una prueba donde interactué con las aplicaciones ya existentes de la AFORE.

El aspecto más preocupante en cuanto a la interacción del STA con el resto del sistema AFORE es el consumo de recursos y la concurrencia, que ya contempló la prueba unitaria y es solventado por el CICS.

CASO	USUARIO SIN AUTORIDAD CICS	USUARIO CON AUTORIDAD APLICATIVA	ACCION IGUAL A 'E', 'R' O ESPACIOS	ACCION IGUAL A ESPACIOS	ACCION IGUAL A 'E'	SELECCION VACIA	SELECCION DE MAS DE UN ARCHIVO O ACUSE	ACCION IGUAL A 'R'	SELECCION DE MAS DE UN ARCHIVO O ACUSE	SELECCION VACIA	RECIBIR ACUSE	ENVIAR ACUSE	RECIBIR ARCHIVO	ENVIAR ARCHIVO	RESULTADO
1	SI														SIN ACCESO AL CICS
2	NO	NO													SIN ACCESO AL FRONT DE TRANSMISION
3		SI	NO												OPCION INVALIDA
4			SI	NO											ACCION IGUAL A 'R' O 'E'; PRIMERO CONSULTE.
5				SI	NO										CONSULTA EFECTUADA
6					SI	SI									SELECCIONE UN ARCHIVO
7						NO	SI								SELECCIONE SOLO UN ARCHIVO
8							NO	SI							TRANSMISION EFECTUADA
9								SI	SI						SELECCIONE SOLO UN ARCHIVO
10									NO	SI					SELECCIONE UN ARCHIVO
11										NO	SI				TRANSMISION EFECTUADA
12											NO	SI			UN ACUSE NO SE PUEDE ENVIAR, SOLO RECIBIR
13												NO	SI		UN ARCHIVO NO SE PUEDE RECIBIR, SOLO ENVIAR
14													NO	SI	TRANSMISION EFECTUADA



Liberación y mantenimiento.

Para poder hacer uso del sistema se tiene que verificar la infraestructura de red, los nodos entre las diferentes organizaciones a intercambiar archivos, revisar el cableado entre computadoras y verificar que este correctamente configurado el protocolo TCP/IP en cada una de las terminales que utiliza el sistema. El equipo más importante sin duda es el HOST. De este equipo es necesario comprobar su disponibilidad en todo momento que sea requerido, así como establecer políticas para su mantenimiento continuo, principalmente en el manejo de la información como los respaldos.

El HOST soporta los usuarios concurrentes para aplicaciones en línea (gracias al CICS) y administra el crecimiento de la base de datos (debido al DB2).

La instalación en una plataforma Mainframe corre a cargo de la utilería denominada Info V6, que es una aplicación encargada de gestionar las versiones de los diferentes componentes (programas, JCL'S, tarjetas, Copys, etc.).

Existen trámites muy variados para liberar un componente en Mainframe, depende directamente de la institución de la que se esta hablando.

Asignación de Roles.

Es importante saber que personas tendrán acceso al sistema antes de ponerlo en marcha, como se ha mencionado, el STA en su primera versión sólo permite dar o no autorización a los usuarios para poder enviar o recibir archivos.

Capacitación y soporte técnico.

Si dentro de la organización que utilizara el sistema no existe personal con los conocimientos adecuados para el mantenimiento del sistema es necesario llevar a cabo cursos de capacitación de varias sesiones semanales para que en el futuro sea personal del cliente quien administre el STA.

Por otro lado, debido a que el STA es amigable, cualquier usuario de CICS dentro de la organización, es capaz de operarlo sólo siguiendo con exactitud el manual de usuario.

El soporte técnico se proporciona en todo momento que el sistema lo requiera por el periodo de tiempo acordado previamente.

Con posterioridad a la fase de implementación de los sistemas, se impone la fase de mantenimiento.

El mantenimiento de sistemas es el mantenimiento continuo después del inicio del funcionamiento.

Cuando se elaboran planes para la estrategia de información, las organizaciones no pueden dejar de considerar que el mantenimiento de sistemas es la fase más prolongada y costosa del ciclo de vida de los sistemas. Las implicaciones del volumen de trabajo para mantenimiento para los planes de estrategia de información en una organización es un tema que merece atención especial. La estructura de organización necesita flexibilidad para



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

apoyar el mantenimiento de los sistemas existentes concurrentemente con la ejecución de nuevas tecnologías.

Es importante considerar la evaluación y el monitoreo de un sistema en términos del mantenimiento necesario y, en consecuencia, reducir o contener los costos implícitos. El mantenimiento de sistemas puede clasificarse en cuatro grupos, cada uno de los cuales repercute en el plan estratégico de información institucional de diferentes maneras:

- Mantenimiento correctivo. Independientemente de cuán bien diseñado, desarrollado y probado está un sistema o aplicación, ocurrirán errores inevitablemente. Este tipo de mantenimiento se relaciona con la solución o la corrección de problemas del sistema. Atañe generalmente a problemas no identificados durante la fase de ejecución. Un ejemplo de mantenimiento correctivo es la falta de una característica requerida por el usuario, o su funcionamiento defectuoso.
- Mantenimiento para fines específicos. Este tipo de mantenimiento se refiere a la creación de características nuevas o a la adaptación de las existentes según lo requieren los cambios en la organización o los usuarios, por ejemplo, los cambios en el código tributario o los reglamentos internos de la organización.
- Mantenimiento para mejoras. Se trata de la extensión o el mejoramiento del desempeño del sistema, ya sea mediante el agregado de nuevas características, o el cambio de las existentes. Un ejemplo de este tipo de mantenimiento es la conversión de los sistemas de texto a GUI (interfaz gráfica de usuarios).
- Mantenimiento preventivo. Este tipo de mantenimiento es probablemente uno de los más eficaces en función de los costos, ya que si se realiza de manera oportuna y adecuada, puede evitar serios problemas en el sistema. Un ejemplo de este mantenimiento es la corrección del problema del año 2000.

4.3 Consideraciones para aumentar la funcionalidad.

Es importante mantener una constante comunicación con los usuarios para conocer sus impresiones sobre el sistema y reunir aquellos aspectos que les gustaría mejorar para incluirlos en las versiones futuras, de tal forma que la entrega del producto no cierre definitivamente las opciones de proveer más y mejores sistemas de información.

Seguridad.

A corto plazo se puede pensar en mejorar la seguridad con la que opera el STA, como se ha mencionado, el sistema permite o no la operación de un usuario, es conveniente, contar en la siguiente versión, con un manejo de seguridad por niveles, es decir, no sólo tener o no acceso al sistema, sino también determinar quien puede o no enviar un archivo en particular o quien puede o no recibir datos, también se puede discriminar por origen y destino de la información.



Funcionalidad.

También se puede considerar a corto plazo la manera en que el usuario puede enviar o transmitir un archivo. Actualmente el usuario puede transmitir un archivo solo si el nombre de éste, la referencia de su ruta física y demás datos referentes al mismo, están almacenados en la entidad TAF239; sería muy conveniente que el usuario pudiera en línea capturar un nuevo archivo a enviar o recibir, que el proceso antes de tratar de transmitirlo, lo agregara a la tabla TAF239 y una vez hecho esto, se efectuara la transmisión.

Otro aspecto que debe considerarse es la manera en que el usuario podría trabajar con un horario automatizado, es decir, con recordatorios de envío o recepción de archivos a ciertas horas preestablecidas por él y administradas por el STA.

4.4 Consideraciones para migrar el sistema a otros equipos.

A medida de que pasa el tiempo y los sistemas evolucionan es necesario aprovechar las ventajas de nuevas tecnologías. La informática y todas las ciencias de la computación son especialmente susceptibles a este evento. En algún momento será necesario utilizar una plataforma que ofrezca la misma prestación que la que se utiliza actualmente (tal vez a un menor costo y sin sacrificar mucho).

El STA se puede desarrollar e implementar lo mismo en un Mainframe que en un servidor NT, por ejemplo, el monitor de transacciones CICS se puede emular bajo la plataforma de Microsoft .NET, el DB2 puede ser sustituido por el manejador de base de datos SQL SERVER y los procesos Batch del Mainframe pueden ser simulados por servicios Windows, asimismo, el sistema operativo Windows brinda (por ser del mismo fabricante) gran soporte a todas las herramientas descritas.

El servidor NT puede proveer a las terminales de los servicios necesarios para utilizar pantallas de tipo Windows desarrolladas en JAVA o en Visual Basic que sirvan de interfaz como Front de transmisión y puedan invocar el servicio que disparara una transmisión utilizando el producto NDM Direct Connect de Sterling Commerce Corp. también disponible para Windows.

Las capacidades de manejo de concurrencia y de ejecución de servicios que proporciona el .NET son realmente sorprendentes si se considera que no se trata de un Mainframe y esta característica ha posicionado a Microsoft como una seria alternativa de procesamiento de un importante volumen de datos.

Si se piensa en el STA como un sistema distribuido en lugar de centralizado (como lo es para el HOST) también se tiene las posibilidades de llevarlo a una arquitectura de Intranet. Basta con ejecutar desde un ambiente Web el servicio ligado a la transacción que realiza el llamado al proceso Batch que efectúa la transmisión. Esta conexión se logra bajo una arquitectura Intranet denominado ICA o bien bajo el precepto cliente servidor PS9 basado en mensajes y eventos.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

El ciclo de pruebas es esencial para determinar si el sistema puede o no ser liberado. Es importante conocer en que medida el sistema esta listo para operar, no existen los términos medios, simplemente esta o no listo.

Una vez liberado se debe contar con una estrategia de mantenimiento, por requerimiento o preventivo, aunque obviamente, por más pruebas que se apliquen, siempre existirá el mantenimiento correctivo.

La continua comunicación con el usuario permite la retroalimentación necesaria para aumentar las capacidades del sistema, para planear mejoras futuras y para su optimización general.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

CONCLUSIONES



CONCLUSIONES.

En las instituciones bancarias la información cambia día a día y por esta razón es importante proporcionar la herramienta adecuada para que los datos que fluyen en todas sus transacciones sean distribuidos a las dependencias gubernamentales que así lo requieran.

La necesidad de investigar las condiciones actuales del negocio y la búsqueda de la solución, de sus objetivos y sus metas fueron parte de la primera etapa de este proceso de desarrollo.

Por otro lado, la adecuada planeación y análisis fue la base del éxito de este sistema. Debido a ello es importante utilizar la metodología adecuada, ya que, esta es fundamental porque proporcionó la guía y dirección para el desarrollo del mismo.

Un sistema de información es valioso cuando se adapta a los cambios de la organización para los que fue creado. La estandarización de código y procesos es sin duda, la base para lograr esta adaptación sin mayor esfuerzo.

El proyecto aquí mostrado es una aplicación hecha a la medida de las necesidades de las instituciones financieras, sin embargo, puede ser tomado en cuenta no sólo para crear aplicaciones de transmisión de información, sino también, para cualquier aplicación de propósito general y para cualquier modelo de negocio.

El STA es una herramienta de software que utilizado adecuadamente permite distribuir en forma y contenido la información que las AFORES (o cualquier otro grupo financiero) producen hacia las dependencias que así lo requieren, de esta forma se logra que la institución que lo emplee ocupe un alto grado de competitividad en el mercado, debido a que la información que intercambie siempre será expedita, rápida, específica, oportuna y sobre todo confiable.

Definitivamente la operación de una AFORE en cuanto a la distribución de información se refiere, se optimiza en gran medida debido al uso del STA.

Se ha tomado en cuenta el ciclo de vida de los sistemas para lograr la integración del STA al resto de la aplicación AFORE.

No cabe duda que empleando con exactitud la metodología para el desarrollo de sistemas (ciclo de vida del software) los resultados finales prácticamente resuelven el problema.

El STA es escalable a las necesidades del negocio, es decir, para un modelo de negocio tan cambiante como lo es una AFORE, el proceso de transmisión de datos, siempre será independiente de los datos que se quieren transmitir.

Finalmente se puede asegurar que el STA cumple con los objetivos planteados al principio del trabajo, al resolver de una manera eficiente la problemática presentada.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

GLOSARIO



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VA NDM SOBRE
PLATAFORMA MAINFRAME”

GLOSARIO.

BATCH:	Vertiente de programación. Un proceso Batch es el que generalmente no necesita interactuar con el usuario y se ejecuta en las organizaciones por la noche para procesar su operación diaria.
CICS:	Customer Information Control System, monitor de transacciones que se utiliza en plataforma Mainframe. Administra la concurrencia y los recursos.
DB2:	Manejador de Base de Datos relacional de IBM usado generalmente en plataforma Mainframe.
HOST:	Computadora anfitriona de las terminales interconectadas a ella para recibir recursos y servicios. Se llama HOST a las Computadoras de IBM conocidas como Mainframe.
JCL:	Job Control Lenguaje. El lenguaje de control de trabajos es un proceso conformado por una serie de pasos que tiene una tarea específica dentro del sistema operativo MVS.
MAINFRAME:	Supercomputadora de IBM capaz de procesar grandes volúmenes de información en muy poco tiempo y sin cometer errores.
NDM:	Producto para transmisión de datos de la compañía Sterling Commerce basado en el protocolo de TCP/IP.
ON-LINE:	Son los procesos que se ejecutan en línea.
OS/390:	Sistema operativo de IBM que se ejecuta en los equipos Mainframe.
QUEUE:	Estructura de datos del CICS que permite viajar información entre un entorno y otro (como el CICS y el TSO).
RACF:	Método de seguridad nativo del CICS. Sistema con que se autentican los usuarios que accedan al CICS.
STA:	Sistema de transmisión de archivos.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VA NDM SOBRE
PLATAFORMA MAINFRAME”**

TSO: Ambiente y utilería que se ejecuta bajo el sistema operativo MVS, dentro de el se puede acceder a una sesión de base de datos o bien a la consola desde donde se ejecutan JCL's.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

BIBLIOGRAFÍA



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

BIBLIOGRAFÍA

Murach's CICS for the COBOL programmer. Raúl Menéndez. 2001, *IEEE Transactions on computers*. IEEE Computer Society. 1998, vol 47. Los Alamitos (Ca): IEEE Computer Society, 1988. ISSN 0018-9340.

DB2/SQL. Tim Martyn. 1989, *McGraw-Hill Book Company*. 1989, 1221 Avenue of the Americas NY. ISBN 0-07-040666-9.

Murach's MVS Tso: Concepts and Ispf (MVS TSO). Doug Lowe. 2001, *IEEE Transactions on computers*. IEEE Computer Society. 1998, vol 47. Los Alamitos (Ca): IEEE Computer Society, 1988. ISSN 0068-9567.

Software Engineering (7th Edition) (International Computer Science Series). Ian Somerville. 2001,



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

RESUMEN, TABLAS Y FIGURAS



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

RESUMEN.

Existe en las AFORES el problema de que el usuario no puede transmitir por sí mismo los archivos a las instituciones gubernamentales como CONSAR, depende del proceso Batch que se ejecuta por la noche o bien del que personal de sistemas tenga tiempo para operar la compleja utilería del sistema para enviar archivos en línea.

Se propone el desarrollo de un sistema que resuelva esta problemática (el STA), este será amigable y le permitirá al usuario independizarse de los ya tan burocráticos procedimientos que actualmente existen para la transmisión de archivos diaria.

El sistema se desarrolla en plataforma Mainframe usando el lenguaje de programación COBOL, el manejador de Bases de Datos DB2, el monitor de transacciones CICS y el NDM como utilería basada en TCP/IP para la transmisión de datos, el STA funciona de la siguiente manera:

- Existe dos módulos: El Batch y el Línea.
- La parte línea corresponde a la interfaz donde el usuario consulta los archivos a transmitir.
- El usuario identifica cual es el archivo a enviar o recibir, lo selecciona, escoge la opción enviar o recibir, le asigna una fecha de operación y finalmente, cuando esta todo listo, presiona <Enter> para realizar la transmisión.
- En ese momento, todos los parámetros (Fecha, acción, archivo) se almacenan en una tabla de Base de Datos TAF239 para que posteriormente el Batch los recupere.
- Además, la parte línea escribe en una QUEUE el nombre del proceso Batch (JCL) que se encargara con todos sus pasos de recuperar los parámetros almacenados en la TAF239.
- El sistema operativo interpreta la QUEUE e invoca el JCL indicado ahí, en ese instante se ejecuta el proceso Batch con todos sus pasos.
- El primer paso verifica que la Base de datos este activa.
- El segundo paso ejecuta el programa BAFPM72 encargado de recuperar los parámetros de la TAF239 (almacenados por la línea) y armar la tarjeta de transmisión NDM a partir de ellos.
- El tercer paso es precisamente la llamada a la utilería NDM para transmitir el archivo que el usuario selecciono.
- El cuarto paso es un programa retardador, su finalidad es hacer tiempo mientras se completa la transmisión para que el siguiente paso recupere la respuesta de la consola y determinar el estatus de la transmisión.
- Este paso es precisamente, la recuperación de la consola de transmisión para verificar el resultado.
- El siguiente paso es enviar la consola al servidor NT donde el usuario puede verificar el estatus de la transmisión.



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
 FACULTAD DE ESTUDIOS SUPERIORES ARAGON
 “SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
 PLATAFORMA MAINFRAME”**

- Finalmente se depura la entidad donde se almacenan los datos de los archivos a transmitir TAF239.

Es así como a grandes rasgos funciona el STA, cumpliendo de esa manera el objetivo de proporcionarle al usuario una herramienta amigable para que pueda transmitir los archivos a los diferentes organismos que los solicitan.

La siguiente tabla muestra el diccionario de Datos mencionado en el Capítulo 3 y que corresponde a la tabla TAF239:

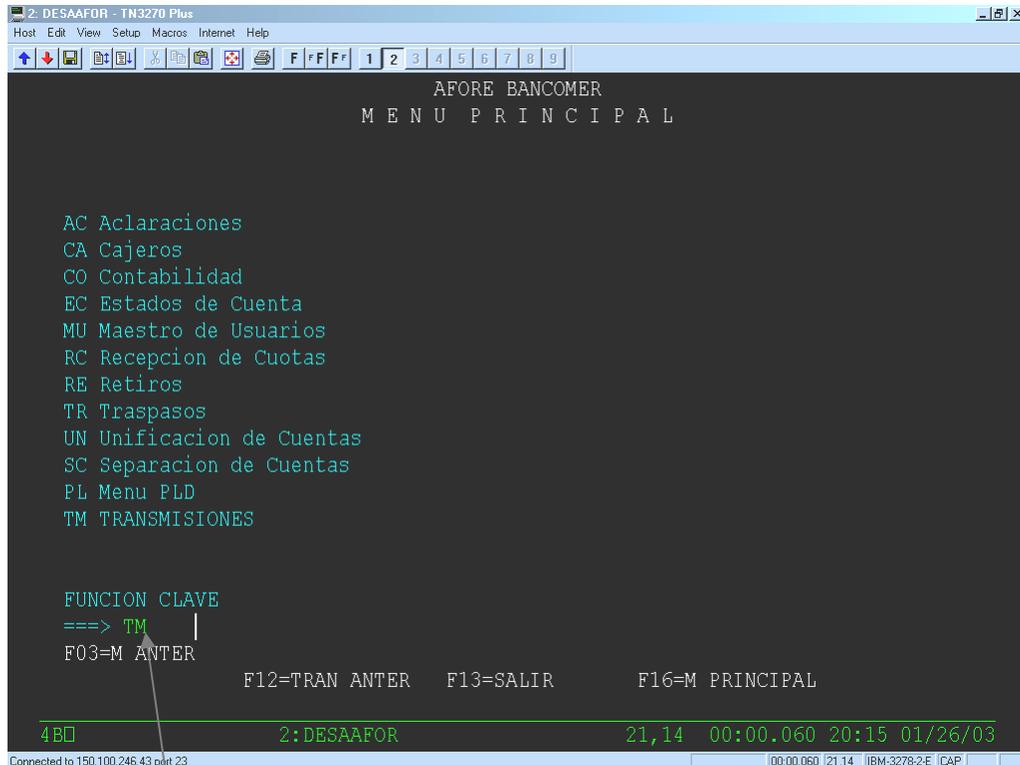
NOMBRE TABLA	Llaves			Nombre de Columna	Tipo	Longitud	Escala	Descripción	Valor Dominio	
DESCRIPCIÓN	PK	FK	FK (Tabla)						Clave	Valor
TAF239_TRANSAR	X		X	IDARCHIVO	char	3		Identificador del Archivo	001-999	
			X	DESCARCH	char	40		Descripción del Archivo		
Tabla que contiene los archivos a Transmírtirse a la CONSAR y sus rutas asociadas.	X		X	ACCION	char	1		Acción Ejecutada	E	Envío de Archivos
									R	Recepción de Archivos
			X	RUTAHOST	char	50		Ruta del Archivo en HOST		
			X	SERVENVIO	char	60		Ruta del Servidor UNIX CONSAR (Para envíos)		
			X	SERVRECEP	char	60		Ruta del Servidor UNIX CONSAR (Para Recepción)		
			X	NOMARCHIVO	char	8		Nombre del Archivo		
	X		X	STATUSARCH	char	2		Estatus actual del Archivo	CA	Rutas catalogadas
									TM	Archivo a Transmírtirse
	X		X	FHACTUAL	date	10		Fecha de Inserción del Reg.		
			X	HMACTUAL	time	8		Hora de Inserción del Reg.		
			X	CDUSUARIO	char	8		Programa que realiza la inserción		



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Manual de Operación del STA.

Para acceder al submenú de transmisiones, dentro del menú de la afore af00:

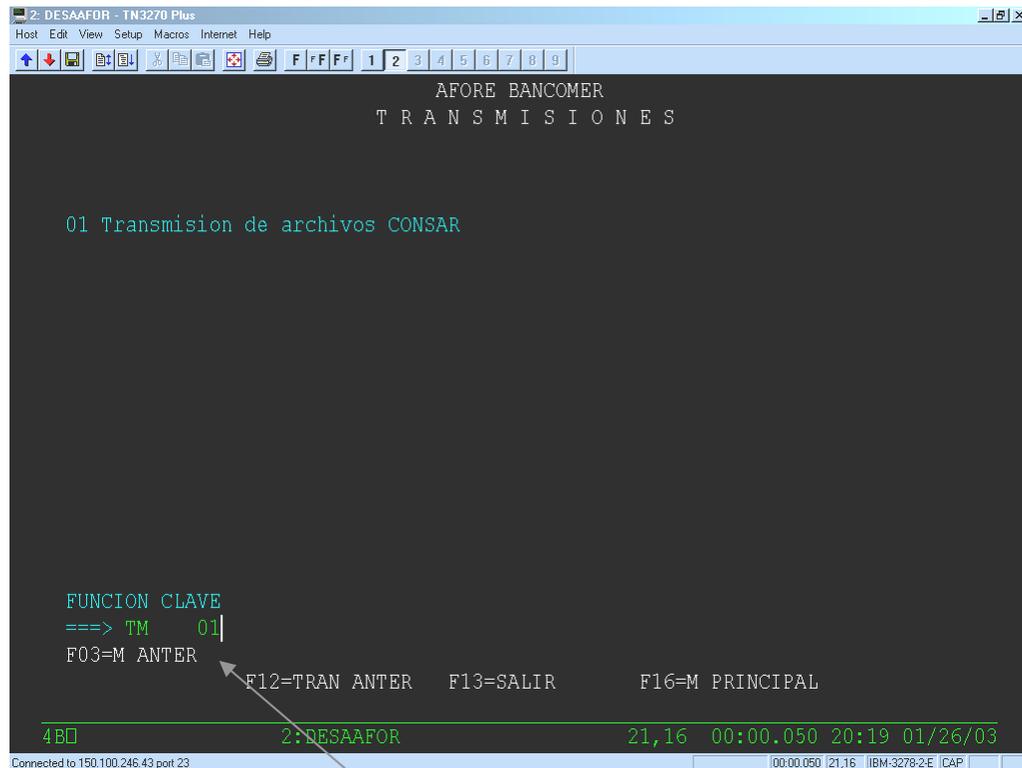


Se tecldea: “TM”.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Una vez hecho lo anterior, se llega al submenú de transmisiones.



Aquí, se elige la única opción disponible en este momento, “transmisión de archivos CONSAR” tecleando “01”.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

Así se entra a la pantalla donde se elige el archivo a enviar, o el acuse a recibir.

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
F F F F 1 2 3 4 5 6 7 8 9
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CON SAR

Accion...ϕ | !          " " = Consulta  E = Enviar  R = Recibir
Archivo: ϕ | !          Fecha (Nombre archivo) ϕ / / !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
ϕ | !
Funcion Clave
===> TM
F03=M ANTER  F06=CANCELA  F07=PAG ATRAS  F08=PAG ADEL
F12=TRAN ANTER  F13=SALIR  F16=M PRINCIPAL

4B□          2:DESAAFOR          04,14 00:00.060 20:23 01/26/03
Connected to 150.100.246.43 port 23          00:00.060 04,14 IBM-3278-2-E CAP
```

Esta pantalla permite:

- Consultar archivos y acuses
- Enviar archivos
- Recibir acuses



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"**

Una vez que se le informa el nombre de archivo (fecha), permite acceder a la consulta de archivos:

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion...  | !
Archivo:  | !
  | 001      " " = Consulta  E = Enviar  R = Recibir
  | 001      Fecha (Nombre archivo)  2003 / 01 / 24 !
  | 014      AFILIACION SEMANAL
  | 035      ACUSE GENERADO POR CONSAR 001
  | 035      BOLETIN
  | 038      REPORTE DE PROMOTORES (QUINCENAL)
  | 038      ACUSE GENERADO POR CONSAR 035
  | 041      REPORTE DE PROMOTORES (MENSUAL)
  | 041      ACUSE GENERADO POR CONSAR 038
  | 043      ADMON. CTAS. INDIV. DE AFORE
  | 043      ACUSE GENERADO POR CONSAR 104
  | 061      ADMON DE CTAS INDIV PREST DE SERVICIOS
  | 061      ACUSE GENERADO POR CONSAR 043
  | 062      BALANZA DE AFORE
  | 062      ACUSE GENERADO POR CONSAR 061
  |          BALANZA DE SIEFORE 1

Funcion Clave
===> TM
F03=M ANTER   F06=CANCELA   F07=PAG ATRAS   F08=PAG ADEL
F12=TRAN ANTER F13=SALIR      F16=M PRINCIPAL

4B  2:DESAAFOR 04.14 00:00.060 20:47 01/26/03
Start | [Icons] | DOCUM... | FASE I | 2: DE... | Microsof... | MUSIC... | 8:47 PM
```



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

De la lista de archivos se elige el que se desea enviar con una "X" (no se puede enviar un acuse ni recibir un archivo) y el campo de acción debe ser una "E" para realizar el envío.

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion...  E !
Archivo:  !
Fecha (Nombre archivo)  2003 / 01 / 24 !
AFILIACION SEMANAL
ACUSE GENERADO POR CONSAR 001
BOLETIN
REPORTE DE PROMOTORES (QUINCENAL)
ACUSE GENERADO POR CONSAR 035
REPORTE DE PROMOTORES (MENSUAL)
ACUSE GENERADO POR CONSAR 038
ADMN. CTAS. INDIV. DE AFORE
ACUSE GENERADO POR CONSAR 104
ADMN DE CTAS INDIV PREST DE SERVICIOS
ACUSE GENERADO POR CONSAR 043
X ! 061
ACUSE GENERADO POR CONSAR 061
! 062
BALANZA DE SIEFORE 1

Funcion Clave
==> TM
F03=M ANTER    F06=CANCELA    F07=PAG ATRAS    F08=PAG ADEL
F12=TRAN ANTER F13=SALIR      F16=M PRINCIPAL

4B  2:DESAAFOR 05.14 00:00.070 20:55 01/26/03
Start | DOCUM... | FASE I | 2: DE... | Microsoft... | New Ord... | 8:55 PM
```



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"**

Con estas condiciones y pulsando <Enter> se efectúa la transmisión, (en caso de que se presente el mensaje "transmisión no efectuada, error en jcl", avisar a sistemas).

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
[Icons] F r F Fr 1 2 3 4 5 6 7 8 9

AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion...  E !
Archivo:  !
" " = Consulta E = Enviar R = Recibir
Fecha (Nombre archivo)  2003 / 01 / 24 !
AFILIACION SEMANAL
ACUSE GENERADO POR CONSAR 001
BOLETIN
REPORTE DE PROMOTORES (QUINCENAL)
ACUSE GENERADO POR CONSAR 035
REPORTE DE PROMOTORES (MENSUAL)
ACUSE GENERADO POR CONSAR 038
ADMN. CTAS. INDIV. DE AFORE
ACUSE GENERADO POR CONSAR 104
ADMN DE CTAS INDIV PREST DE SERVICIOS
ACUSE GENERADO POR CONSAR 043
BALANZA DE AFORE
ACUSE GENERADO POR CONSAR 061
BALANZA DE SIEFORE 1

Funcion Clave
==> TM
F03=M ANTER      F06=CANCELA      F07=PAG ATRAS      F08=PAG ADEL
                  F12=TRAN ANTER      F13=SALIR          F16=M PRINCIPAL

TRANSMISION EFECTUADA
4B 2:DESAAFOR 04,14 00:00.060 20:59 01/26/03
Connected to 150.100.246.43 port 23 00:00.060 04,14 IBM-3278-2-E ICAP
```



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

Ahora se hará una recepción de acuse.

Se Recuperará el acuse del archivo que se envió en el ejemplo anterior: "balanza de siefore 1", después del tiempo pertinente.

Los pasos son los mismos que los del envío, solo cambian las condiciones; también se debe seleccionar con "X" el acuse (no se puede recibir archivos ni enviar acuses) y en el campo acción teclear una "R".

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CON SAR

Accion...  R !
Archivo:  !
  ! 062      " " = Consulta  E = Enviar  R = Recibir
  X ! 062      Fecha (Nombre archivo)  2003 / 01 / 24 !
  ! 068      BALANZA DE SIEFORE 1
  ! 068      ACUSE GENERADO POR CON SAR 062
  ! 071      APORTACIONES VOLUNTARIAS
  ! 071      ACUSE GENERADO POR CON SAR 068
  ! 080      TENENCIA Y OBTENCION DE PRECIO SIEFORE
  ! 080      ACUSE GENERADO POR CON SAR 071
  ! 089      ENVIO DE PROMOTORES
  ! 096      ACUSE GENERADO POR CON SAR 080
  ! 096      ACUSE GENERADO POR CON SAR 089
  ! 099      TRASPASOS
  ! 099      ACUSE GENERADO POR CON SAR 104
  ! 101      TRASPASOS GESTION CEDENTE
  ! 101      ACUSE GENERADO POR CON SAR 099
           TRASPASOS GESTION PRESTADORA

Funcion Clave
===> TM
F03=M ANTER   F06=CANCELA   F07=PAG ATRAS   F08=PAG ADEL
F12=TRAN ANTER F13=SALIR     F16=M PRINCIPAL

4B7          2:DESAAFOR          08.05 00:00.110 21:04 01/26/03
Start | [Icons] | [Taskbar] | 9:04 PM
```



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"**

Si todo es correcto, se efectuará la transmisión (en este caso la recepción de un acuse).

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion... R !
Archivo: !
! 062
! 062
! 068
! 068
! 071
! 071
! 080
! 080
! 089
! 096
! 096
! 099
! 099
! 101

Funcion Clave
==> TM
F03=M ANTER      F06=CANCELA      F07=PAG ATRAS      F08=PAG ADEL
F12=TRAN ANTER   F13=SALIR        F16=M PRINCIPAL

TRANSMISION EFECTUADA
4B  2:DESAAFOR  04.14  00:00.050  21:06 01/26/03
```




**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”**

Ahora se verán los errores a los que el usuario es susceptible, pero que el sistema inteligentemente evita:

Supóngase que el usuario selecciona un acuse y en el campo acción teclea “E”, esto es erróneo ya que, no se deben enviar acuses sino archivos, se verá que el sistema no admite esta acción y sugiere la corrección inmediata de la acción....

```
2: DESAAFOR - TN3270 Plus
Host Edit View Setup Macros Internet Help
F F F F F 1 2 3 4 5 6 7 8 9

AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion...  E !
Archivo:  !
  ! 001
  ! 001
  X ! 014
  ! 035
  ! 035
  ! 038
  ! 038
  ! 041
  ! 041
  ! 043
  ! 043
  ! 061
  ! 061
  ! 062

Funcion Clave
==> TM
F03=M ANTER      F06=CANCELA      F07=PAG ATRAS    F08=PAG ADEL
                  F12=TRAN ANTER   F13=SALIR        F16=M PRINCIPAL

SOLO PUEDE ENVIAR ARCHIVOS, NO ACUSES
4BP 2:DESAAFOR 04.14 00:00.050 21:18 01/26/03
```



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
"SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME"

La misma protección sucede en viceversa.....recibiendo un archivo:

```
2: DESAAFDR - TN3270 Plus
Host Edit View Setup Macros Internet Help
AFORE BANCOMER
TRANSMISION DE ARCHIVOS CONSAR

Accion...!
Archivo: !
! 001 AFILIACION SEMANAL
! 001 ACUSE GENERADO POR CONSAR 001
! 014 BOLETIN
X! 035 REPORTE DE PROMOTORES (QUINCENAL)
! 035 ACUSE GENERADO POR CONSAR 035
! 038 REPORTE DE PROMOTORES (MENSUAL)
! 038 ACUSE GENERADO POR CONSAR 038
! 041 ADMON. CTAS. INDIV. DE AFORE
! 041 ACUSE GENERADO POR CONSAR 104
! 043 ADMON DE CTAS INDIV PREST DE SERVICIOS
! 043 ACUSE GENERADO POR CONSAR 043
! 061 BALANZA DE AFORE
! 061 ACUSE GENERADO POR CONSAR 061
! 062 BALANZA DE SIEFORE 1

Funcion Clave
==> TM
F03=M ANTER F06=CANCELA F07=PAG ATRAS F08=PAG ADEL
F12=TRAN ANTER F13=SALIR F16=M PRINCIPAL

SOLO PUEDE RECIBIR ACUSES, NO ARCHIVOS
4B 2: DESAAFDR 04.14 00:00.050 21:20 01/26/03
```



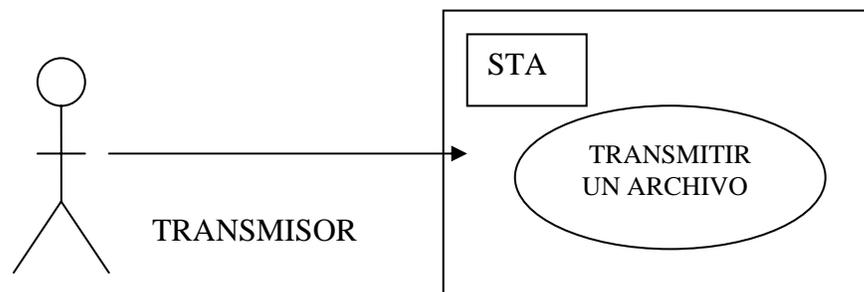
Casos de Uso del STA.

Los casos de uso son descripciones narrativas en el lenguaje natural de los procesos del dominio en un formato estructurado de prosa. Describen una secuencia de acciones.

Caso de uso: Transmitir un archivo.

Participantes: Transmisor.

Descripción: Este caso de uso comienza cuando el transmisor envía un archivo al receptor. El Transmisor es el usuario que se encuentra en la AFORE y envía un archivo a la dependencia gubernamental.

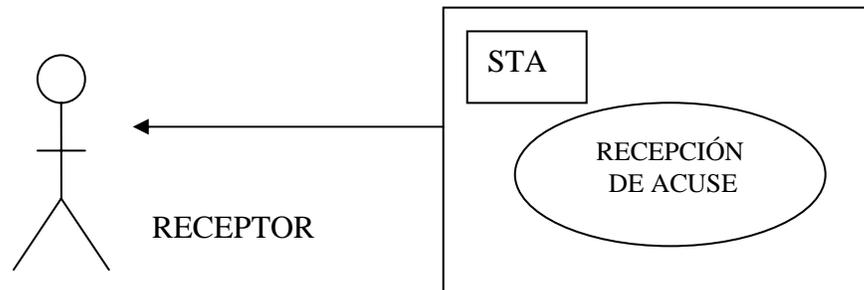




Caso de uso: Recepción de un acuse.

Participantes: Receptor.

Descripción: Este caso de uso comienza cuando el organismo gubernamental envía al usuario de AFORE un acuse de que recibió el archivo que éste le envió. El Receptor es entonces el usuario que se encuentra en la Afore y le es entregado por medio del STA un acuse proveniente de la dependencia gubernamental.

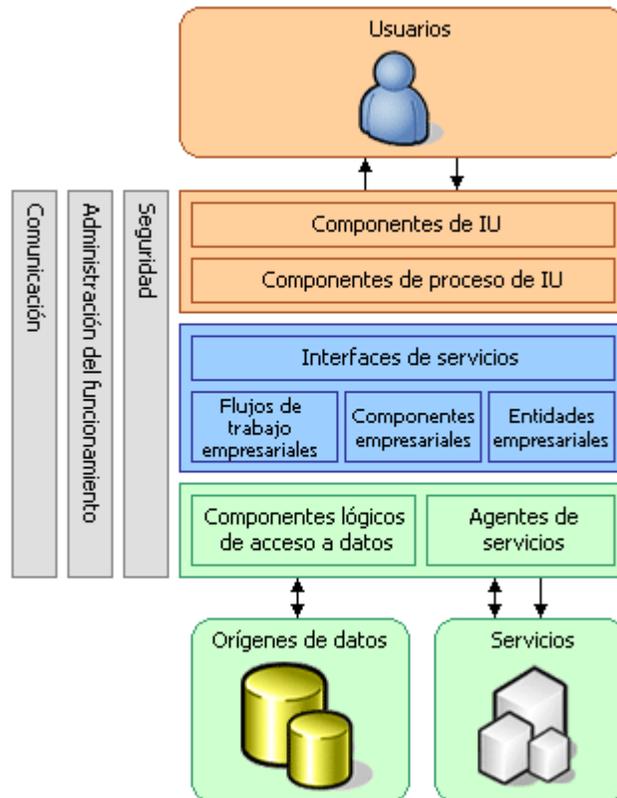




STA: Arquitectura del sistema en tres capas.

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.





UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON
“SISTEMA DE TRANSMISIÓN DE ARCHIVOS VÍA NDM SOBRE
PLATAFORMA MAINFRAME”

Esta potente arquitectura otorga algunas ventajas:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

