


---

# Manejo de Datasets

Roberto Gómez Cárdenas  
rogomez@itesm.mx

Lámina 1 Roberto Gómez C.




# Data Set

---

- Colección de registros de datos relacionados lógicamente y almacenados en un volumen de almacenamiento de disco o un conjunto de volúmenes.
- Un data set puede ser
  - un programa fuente
  - un biblioteca de macros
  - un archivo de registros de datos usado por un programa de procesamiento
- Es posible imprimir un data set o desplegarlo en una terminal.
- El registro lógico es la unidad básica de información usada por un programa corriendo en z/OS

Lámina 2 Roberto Gómez C.




## Los registros

---

- Registro: número fijo de bytes que contiene datos.
- Seguido un registro colecta información relacionada que se trata como una unidad.
  - p.e. un determinado artículo en una base de datos o datos personales de una persona.
  - El término campo se usa para referirse a esta porción de datos.
- Unidad básica de información usada por un programa corriendo en z/OS.
- Pueden ser organizados de diferentes formas dependiendo de cómo se planea acceder a la información.


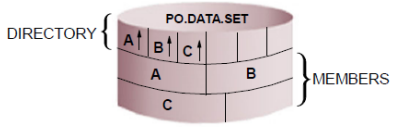
Lámina 3
Roberto Gómez C.



## Tipos data sets

---

- Secuencial
  - registros son almacenados consecutivamente
  - para acceder al decimo se debe pasar por los primeros nueve
- Particionado (PDS)
  - consiste de un directorio y miembros
  - directorio contiene dirección de cada miembro
  - posible acceder a los miembros directamente
  - también son conocidos como librerías
- VSAM
  - registros KSDS (Key Sequenced Data Set) son almacenados con información de control (llaves)
  - se pueden acceder directamente
  - ideales para data sets usados frecuentemente y en orden impredecible

VSAM Data set types:

- KSDS
- ESDS
- LDS
- RRDs
- Fixed Length
- Variable Length

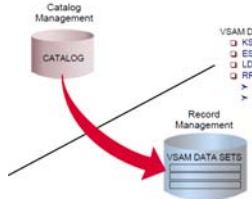



Lámina 4
Roberto Gómez C.



## Almacenamiento datos en z/OS

---

- z/OS soporta diferentes dispositivos para almacenamiento de datos.
- Los más frecuentes:
  - Discos: DASD
  - Cintas
- DASD
  - Direct Access Storage Devices
  - Algunos data sets se almacenan de forma secuencial, sin embargo es posible manejar acceso directo.
- Cintas
  - Conocidas como dispositivos de acceso secuencial ya que deben ser accedidos secuencialmente.

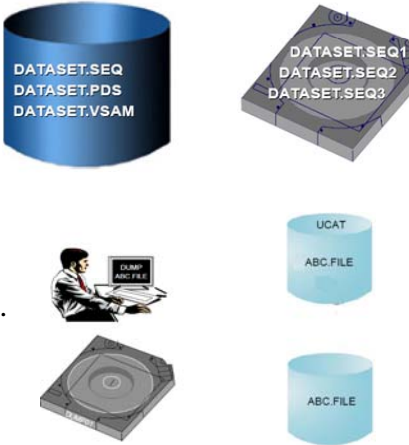


Lámina 5
Roberto Gómez C.




## Los DASD

---

- Direct Access Storage Device
- Posible almacenar y retirar registros, ya sea directamente o secuencialmente
- Se usan volúmenes DASD para el almacenamiento de datos y programas ejecutables, incluyendo el sistema operativo mismo, y para almacenamiento de trabajo temporal.
- Posible usar un volumen DASD para varios data sets y reasignar o reusar espacio en el volumen.

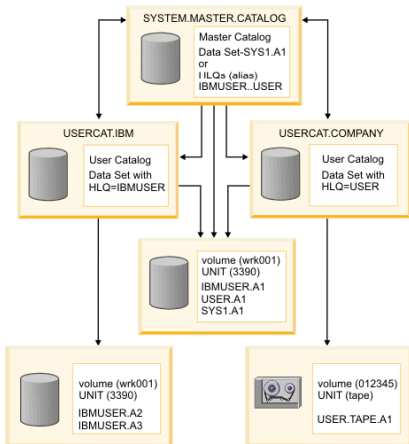
Lámina 6
Roberto Gómez C.



## Localización de data sets

---


- Para localizar un data set de forma rápida:
  - z/OS incluye un data set conocido como *catálogo maestro*.
  - *Cátalogo Maestro* permite acceso a cualquier data set en el sistema o a otros catálogos.
  - Necesario que el catálogo maestro resida en un DASD que siempre se encuentre montado en un drive que se encuentre en línea con el sistema.



El diagrama ilustra la estructura de catálogos y volúmenes en z/OS. En la parte superior, el 'SYSTEM.MASTER.CATALOG' (Master Catalog Data Set-SYS1 A1) actúa como punto central, con alias para H.L.Q.s como IBMUSER.USER. Este se conecta con dos catálogos de usuario: 'USERCAT.IBM' (Data Set with H.L.Q.=IBMUSER) y 'USERCAT.COMPANY' (Data Set with H.L.Q.=USER). Desde estos catálogos, se accede a volúmenes de datos: 'volume (wrk001) UNIT (3390) IBMUSER A1 USER A1 SYS1 A1' y 'volume (wrk001) UNIT (3390) IBMUSER A2 IBMUSER A3'. También se muestra un volumen de cinta: 'volume (012345) UNIT (tape) USER.TAPE A1'.

Lámina 7

Roberto Gómez C.




## Administración datos en z/OS

---

- Involucra las siguientes tareas
  - allocation, placement, monitoring, migration, backup, recall, recovery, and deletion.
- Administración de almacenamiento puede hacerse ya sea manualmente o a través de procesos automáticos (o a través de una combinación de los dos)
- En z/OS, DFSMS es usado para automatizar la administración de almacenamiento de los data sets
  - DFMS: Data Facility Storage Management Subsystem

Lámina 8


Roberto Gómez C.



## ¿Qué es un método de acceso?

- Define la técnica usada para almacenar y recuperar datos.
- Incluye programas proporcionados por el sistema y utilerías que definen y procesan data sets
- Métodos de acceso comúnmente usados incluyen lo siguiente
  - VSAM, QSAM, BSAM, BDAM y BPAM


Lámina 9 Roberto Gómez C.



## Los métodos de acceso

- QSAM: Queued Sequential Access Method
  - muy usado
- BSAM: Basic Sequential Access Method
  - para casos especiales
- BDAM: Basic Direct Access Method
  - se esta volviendo obsoleto
- BPAM: Basic Partitioned Access Method
  - para bibliotecas
- VSAM: Virtual Sequential Access Method
  - usado para aplicaciones más complejas

Lámina 10 Roberto Gómez C.




## Volumenes DASD

---


- Usados para almacenar datos y programas ejecutables, así como para almacenamiento temporal de trabajo.
- En un volumen el nombre de un data set debe ser único.
- Una data set puede ser localizado por el tipo de dispositivo, el número de serie del dispositivo y el nombre del data set.
- La estructura del sistema de archivos de z/OS no es jerárquica.

**DASD volume**



**volser=DASD01**


**tape volume**



**volser=SL0001**

Lámina 11

Roberto Gómez C.



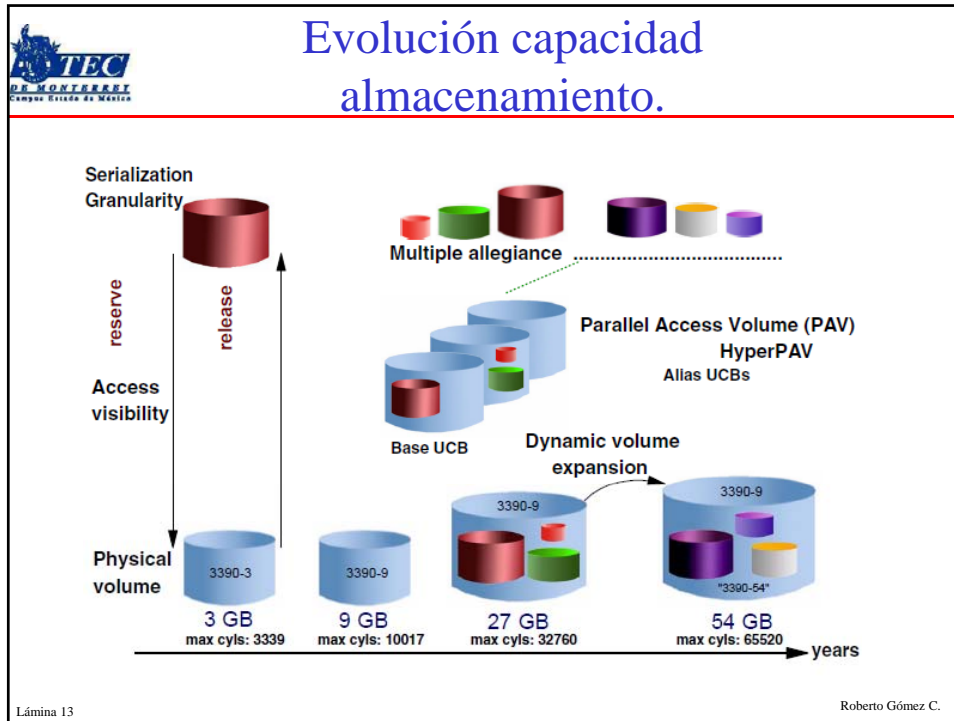
## Terminologías DASD desde el punto de vista de Unix y PCs

---

- DASD es otro nombre para un drive de disco.
- Un drive de disco también se conoce como volumen de disco, o un HDA (Head Disk Assembly).
- Un disk drive contiene cilindros.
- Los cilindros contienen tracks.
- Los tracks contiene registros de datos.
- Los bloques de datos son las unidades de almacenamiento en disco.

Lámina 12


Roberto Gómez C.



### Etiquetas DASD

- Sistema operativo usa grupos de etiquetas para identificar volúmenes DASD y los data sets que contienen.
- Programas aplicación no usan estas etiquetas.
- Volúmenes DASD deben usar etiquetas estándar que incluyen:
  - Etiqueta de volumen.
  - Etiqueta de data set.
  - Opcionalmente etiquetas de usuario.


Lámina 14 Roberto Gómez C.



## Asignando un data set

- Para usar un data set primero debe asignarlo
  - asignarlo = allocate = establecer una liga con él
- La asignación del data set implica:
  - crear espacio para un nuevo *data set* en el disco
  - establecer una liga lógica entre un *job step* y un *data set*
- Después se accede a los datos usando macros relacionados con el método de acceso seleccionado.
- Varias formas para asignar un data set
  - ISPF data set
  - Access Method Services
  - TSO ALLOCATE command
  - job control language (JCL)

Lámina 15 Roberto Gómez C.




## Convención nombres dataset

- Nombre único dentro del volumen.
  - máximo 44 caracteres
  - los puntos forman parte del conteo
- Máximo 22 segmentos de nombre: level qualifier
  - El primer nombre en la izquierda: High Level Qualifier HLQ
  - El último nombre en la derecha: Low Level Qualifier LLQ
- Cada level qualifier
  - De 1 a 8 caracteres.
  - El primer carácter debe ser alfabético o especial (@#%).
  - Los siete restantes: alfabéticos, especiales, numéricos (0-9) o guión (-).

Lámina 16 Roberto Gómez C.






## Ejemplo de nombres no validos

- HLQ.ABCDEFGHI.XYZ
- HLQ..ABC
- HLQ.ABC.
- HLQ.123.XYZ

Lámina 17 Roberto Gómez C.




## Convenciones en el nombre

- HLQ del data set del usuario es controlado por el sistema de seguridad
- Convenciones para el resto (no reglas)
  - letras LIB indican que es una librería
    - letras PDS son una alternativa menos usada para esto
  - letras CNTL, JCL o JOB indican que contiene JCL
  - letras LOAD, LOADLIB o LINKLIB indica que contiene ejecutables
  - letras PROC, PRC o PROCLIB indican una librería o procedimientos JCL
  - Usar muchos qualifiers es considerado una mala práctica
    - P390A.A.B.C.D.E.F.G.H.I.J.K.L.M.N.O.P.Q.R.S

Lámina 18 Roberto Gómez C.






## Registros lógicos y bloques

- LRECL: Logical RECORD Length
  - La porción más pequeña de datos a procesar.
  - Por ejemplo: una cuenta, un empleado, un cliente, etc.
- Registros lógicos ubicados en DASD son agrupados dentro de registros físicos llamados bloques.
  - BLKSIZE indica la longitud de dicho bloque
- Cada bloque cuenta con una ubicación y una dirección única.
- Registros lógicos pueden ser almacenados y retirados ya sea directa o secuencialmente.
- La longitud máxima de un registro lógico, (LRECL) limitado por el tamaño físico del medio usado.


Lámina 21 Roberto Gómez C.



## Ejemplo petición espacio disco

- Cuando la cantidad espacio requerido es expresada en bloques, es necesario especificar el número y longitud promedio de los bloques dentro del data set
- Ejemplo de una petición para almacenamiento en disco:
  - Promedio longitud bloque en bytes: 300
  - Cantidad primaria (número) de bloques : 5,000
  - Cantidad secundaria de bloques, a ser asignados si la cantidad primaria se llena con datos: 100
- A partir de esta información el sistema operativo estima y asigna la cantidad de espacio en disco requerido.


Lámina 22 Roberto Gómez C.



## Extents del data set

- El espacio en disco para para data sets es asignado en *extents*.
- Un *extent* es un número contiguo de tracks, cilindros o bloques.
- Data sets pueden incrementarse en *extents* conforme crecen.
- Viejos tipos de data sets podían contar con 16 extents por volumen.
  - nuevos tipos pueden soportar hasta 128 extents por volumen o 255 extents en diferentes volúmenes.
- Útiles cuando no se están usando PDSEs y se esta manejando el espacio de forma manual (no a través de DFSMS)


Lámina 23 Roberto Gómez C.



## Formato data sets

- Data sets son orientados registros
  - no son orientados byte stream como en PC y sistemas Unix
- No cuentan con caracteres como NL (New Line), RET o line feed (CR+LF) para designar el final del registro.
- Registros son de longitud fija o variable
  - si se edita un data set con ISPF, cada línea es un registro
- Cuentan con cinco formatos
  - un bloque es lo que se escribe en disco
  - un registro es una entidad lógica


Lámina 24 Roberto Gómez C.



## Formatos Data Sets

- **F - Fixed**
  - Registro lógico = bloque físico
  - Todos son del mismo tamaño
  - Raramente usado
- **FB Fixed Blocked**
  - Varios registros lógicos dentro de uno físico.
  - Proporciona eficiente uso espacio y operación.
  - Usado para registros de tamaño fijo.
- **V Variable**
  - cada registro lógico es un bloque físico.
  - La longitud de los registros puede ser de longitud variable.
  - Cada registro es precedido de un RDW (Record Descriptor Word) que describe la longitud de dicho registro.


Lámina 25 Roberto Gómez C.



## Formatos Data Sets

- **VB - Variable Blocked**
  - Consiste de varios registros de longitud variable, cada uno con su propio RDW dentro de un bloque físico.
  - Se cuentan con un BDW (Block Descriptor Word) al principio del bloque, conteniendo la longitud total del bloque.
- **Undefined**
  - Consiste de varios registros/bloques de longitud variable sin estructura predefinida
  - Permite procesamiento de registros que no se acopla al formato V o F.
  - Usado solo para módulos ejecutables

Lámina 26 Roberto Gómez C.



## Los cinco formatos de Data Sets

---

Fixed

F

record

record

record

record

Fixed records.

Fixed Blocked

FB

record

record

record

record

record

record

Fixed blocked records.  $BLKSIZE = n * LRECL$

Variable

V

record

record

record

Variable records.

Variable Blocked

VB

record

record

record

record

record

Variable blocked records.  $BLKSIZE \geq 4 + n * \text{largest } LRECL$

Undefined

U

record

record


record

record

Undefined records. No defined internal structure for access method.

$RDW = \text{Record Descriptor Word} = [ \text{long reg} + 4\text{bytes RDW} ]$

Lámina 27
Roberto Gómez C.




## Ejemplo Data Set FB

---

- Data set con RECFM=FB y LRECL=25
- Es un data set con un longitud de registro de 25 bytes, la B es por Blocked.
- Para un data set FB, el LRECL indica la longitud de cada registro en el data set; todos los registros son de la misma longitud.
- El primer byte de datos en un registro VB se encuentra en la posición 1.
- Un registro en un data set FB con LRECL=25 puede verse como sigue:

Positions 1-3: Country Code = 'USA'  
 Positions 4-5: State Code = 'CA'  
 Positions 6-25: City = 'San Jose' padded with 12 blanks on the right

Lámina 28
Roberto Gómez C.




## Ejemplo Data Set VB

---

- Data set con RECFM=VB y LRECL=25
- Es un data set con un longitud de registro variable, máxima de 25 bytes, la V es de Variable.
- En un data set VB, los registros pueden tener longitud variable.
  - Los primeros cuatro bytes de cada registro contienen el RDW
  - Los primero dos bytes del RDW contienen la longitud del registro (en binario)
- El primer byte de datos en un registro FB se encuentra en la posición 5, después de los 4 bytes RDW en posiciones 1-4.
- Un registro en un data set VB con LRECL=25 puede verse como sigue:
 

Positions 1-2: Length in RDW = hex 0011 = decimal 17  
 Positions 3-4: Zeros in RDW = hex 0000 = decimal 0  
 Positions 5-7: Country Code = 'USA'  
 Positions 8-9: State Code = 'CA'  
 Positions 10-17: City = 'San Jose'

Lámina 29 Roberto Gómez C.




## Terminología

---

- Record Format (RECFM)
  - Se utilizan las letras: F, FB, V, VB o U
- Block Size (BLKSIZE)
  - Tamaño bloque físico para registros F y FB.
  - Para registros V, VB y U es el máximo tamaño de bloque físico.
- Logical Record Size (LRECL)
  - Tamaño lógico del registro (F, FB) o el máximo tamaño permitido para los registros lógicos (V, FB).
  - Registros U no tienen LRECL

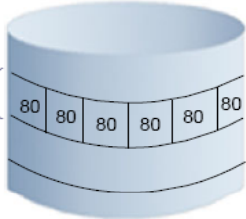
Lámina 30 Roberto Gómez C.



## Esquema Data Set FB

---

Data Set {




DATA SET: TEST: SEQ1

DSORG=PS  
 RECFM=FB  
 LRECL=80  
 BLKSIZE=27920

Lámina 31

Roberto Gómez C.



## Usando data set


---

- Para usar un data set primero debe asignarlo
  - asignarlo = allocate = establecer una liga con él
- La asignación del data set implica:
  - crear espacio para un nuevo *data set* en el disco
  - establecer una liga lógica entre un *job step* y un *data set*
- Después de acceder a los datos usando macros relacionados con el método de acceso seleccionado.
- Varias formas para asignar un data set
  - ISPF data set
  - Access Method Services
  - TSO ALLOCATE command
  - job control language (JCL)

Lámina 32

Roberto Gómez C.






## Métodos asignación data sets

---

- **ALLOCATE**
  - comando ALLOCATE del TSO para crear data sets
  - comando guía a través valores asignación que se deben especificar
- **Menús ISPF**
  - posible usar ISPF para asignar ubicación data set
- **JCL**
  - uso conjunto comandos llamados Job Control Language para asignar data sets
- **Access method services**
  - comandos para trabajar con data sets
  - ejemplo: ALLOCATE, ALTER, DELETE y PRINT

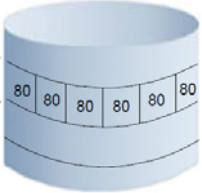
Lámina 33 Roberto Gómez C.



## Asignando espacio en volúmenes DASD a través de JCL

---

- **Necesario especificar**
  - Cantidad espacio requerido
    - En bloques, registros, tracks o cilindros
    - Explícitamente (parámetro SPACE)
    - Implícitamente (SMS data class)



Data Set {

DATASET.TEST.SEQ1

**DSORG=PS**

**RECFM=FB**

**LRECL=80**

**BLKSIZE=27920**


DSORG: Organización del data set Secuencial (PS), particionado (PO) o directo (PA)

Características del registro. Longitud fija (F), longitud variable (V), longitud indefinida (U), longitud variable ASCII (D).

Especifica la longitud del registro en bytes, de cada registro en el data set.

Especifica la longitud máxima en bytes del registro físico (bloque).

Lámina 34 Roberto Gómez C.



## Ejemplo código JCL

---

- Creando un Data Set
 

```
//jobname JOB (start of JOB statement parameters)
//stepname EXEC PGM=IEFBR14
//ddname DD DSN=dsname,
//  DISP=(NEW,CATLG),
//  UNIT=SYSALLDA,SPACE=(TRK,1)
/*
```
- Asignando espacio
 

```
//REPORT DD DSN=LIBRARY.REPORT.FEB08,DISP=(NEW,CATLG),
//          SPACE=(CYL,(1,1)),UNIT=SYSDA,
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=6160)
```

Lámina 35
Roberto Gómez C.




## Asignación en ISPF

---

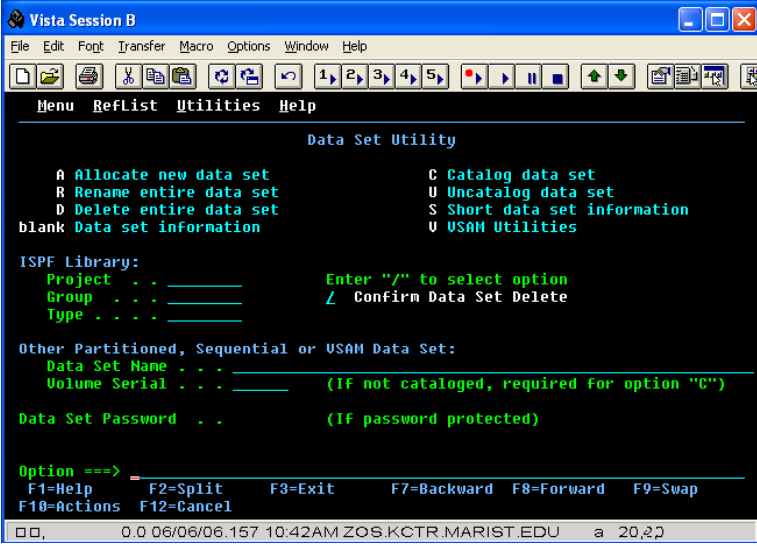


The screenshot shows the 'Utility Selection Panel' in an ISPF environment. The panel lists 15 utilities with their descriptions and associated dialog types. The 'Data Set' utility (number 2) is selected, indicated by a cursor and the text 'Option ==> 2'. The interface includes a menu bar, a toolbar, and a status bar at the bottom showing system information like '0.0 06/06/06.157 10:38AM ZOS.KCTR.MARIST.EDU a 20,40'.

Lámina 36
Roberto Gómez C.



## Utilería Data Set



```

Vista Session B
File Edit Font Transfer Macro Options Window Help
Menu RefList Utilities Help

Data Set Utility

A Allocate new data set          C Catalog data set
R Rename entire data set        U Uncatalog data set
D Delete entire data set        S Short data set information
blank Data set information      U USAM Utilities


ISPF Library:
Project . . . _____      Enter "/" to select option
Group . . . _____        / Confirm Data Set Delete
Type . . . _____

Other Partitioned, Sequential or USAM Data Set:
Data Set Name . . . _____
Volume Serial . . . _____ (IF not cataloged, required for option "C")
Data Set Password . . . _____ (IF password protected)

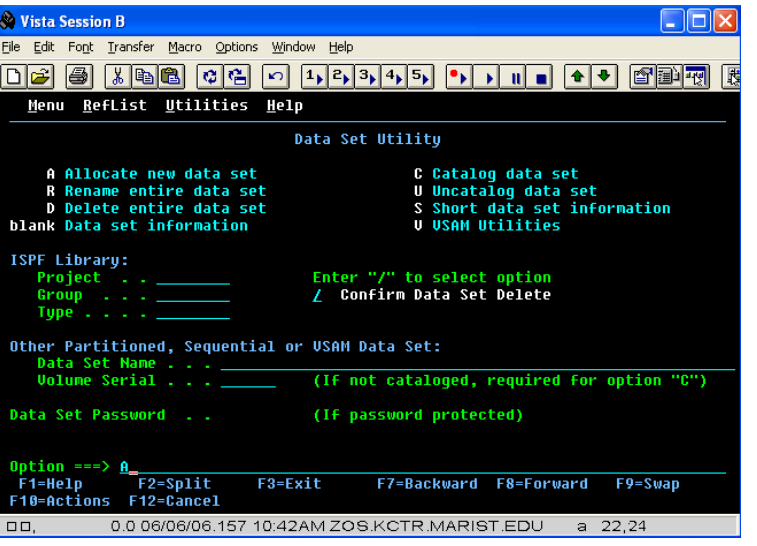
Option ==>
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel

0.0 06/06/06.157 10:42AM ZOS.KCTR.MARIST.EDU a 20,20
    
```

Lámina 37 Roberto Gómez C.



## Opción A: Allocate new Data Set



```

Vista Session B
File Edit Font Transfer Macro Options Window Help
Menu RefList Utilities Help

Data Set Utility

A Allocate new data set          C Catalog data set
R Rename entire data set        U Uncatalog data set
D Delete entire data set        S Short data set information
blank Data set information      U USAM Utilities

ISPF Library:
Project . . . _____      Enter "/" to select option
Group . . . _____        / Confirm Data Set Delete
Type . . . _____

Other Partitioned, Sequential or USAM Data Set:
Data Set Name . . . _____
Volume Serial . . . _____ (IF not cataloged, required for option "C")
Data Set Password . . . _____ (IF password protected)

Option ==> A
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel

0.0 06/06/06.157 10:42AM ZOS.KCTR.MARIST.EDU a 22,24
    
```


Lámina 38 Roberto Gómez C.

**Introduciendo nombre del Data Set**

Lámina 39 Roberto Gómez C.

**Menu Opciones Data Sets**

Lámina 40 Roberto Gómez C.



## Asignando valores del Data Set



```

Vista Session B
File Edit Font Transfer Macro Options Window Help
Allocate New Data Set
Data Set Name . . . : KC03EAD.AREA.CODES
Management class . . . (Blank for default management class)
Storage class . . . (Blank for default storage class)
Volume serial . . . (Blank for system default volume) **
Device type . . . (Generic unit or device address) **
Data class . . . (Blank for default data class)
Space units . . . : TRK (BLKS, TRKS, CYLS, KB, MB, BYTES or RECORDS)
Average record unit (M, K, or U)
Primary quantity . . . 1 (In above units)
Secondary quantity . . . 1 (In above units)
Directory blocks . . . 0 (Zero for sequential data set) *
Record format . . . FB
Record length . . . 80
Block size . . . : 320
Data set name type : LIBRARY (LIBRARY, HFS, PDS, or blank) *
Command ==>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
0.0 06/06/06.157 10:53AM ZOS.KCTR.MARIST.EDU A 2,14
    
```

Lámina 41

Roberto Gómez C.



## Otros posibles valores



```

Vista Session B
File Edit Font Transfer Macro Options Window Help
Allocate New Data Set
Data Set Name . . . : KC03EAD.AREA.CODES
Management class . . . (Blank for default management class)
Storage class . . . (Blank for default storage class)
Volume serial . . . (Blank for system default volume) **
Device type . . . (Generic unit or device address) **
Data class . . . (Blank for default data class)
Space units . . . : TRACK (BLKS, TRKS, CYLS, KB, MB, BYTES or RECORDS)
Average record unit (M, K, or U)
Primary quantity . . . 1 (In above units)
Secondary quantity . . . 1 (In above units)
Directory blocks . . . 0 (Zero for sequential data set) *
Record format . . . FB
Record length . . . 80
Block size . . . : 640
Data set name type : LIBRARY (LIBRARY, HFS, PDS, or blank) *
Command ==>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
0.0 06/06/06.157 10:56AM ZOS.KCTR.MARIST.EDU A 22,24
    
```

Lámina 42

Roberto Gómez C.

### Data Set Asignado

The screenshot shows a terminal window titled "Vista Session B" with a menu for "Data Set Utility". The menu options are:

- A Allocate new data set
- R Rename entire data set
- D Delete entire data set
- blank Data set information
- C Catalog data set
- U Uncatalog data set
- S Short data set information
- U USAM Utilities


Below the menu, there are fields for "ISPF Library:" (Project, Group, Type) and "Other Partitioned, Sequential or USAM Data Set:" (Data Set Name, Volume Serial, Data Set Password). At the bottom, there are function key definitions: F1=Help, F2=Split, F3=Exit, F7=Backward, F8=Forward, F9=Swap, F10=Actions, F12=Cancel. The status bar at the bottom shows "0,4 06/06/06.167 10:58AM ZOS.KCTR.MARIST.EDU a 22,24".

Lámina 43 Roberto Gómez C.

### Ejemplo edición Data Set

The screenshot shows a terminal window titled "Session A - [24 x 80]" in edit mode. The command line is "EDIT ZPROF07.LIB.SOURCE2" and the cursor is at the end of the line. The screen displays a message: "Warning- The UNDO command is not available until you change your edit profile using the command RECOVERY ON." Below this, there is a block of text: "kfkflldkfkfjffkd", "Esto es una prueba de un archivo plano", "Se debe empezar a insertar a partir de la columna 2", "Para volver a editar, al igual que vi, se debe ir a la columna 1 y en modo comando teclear un ix donde x es el numero de lineas a sumar...". At the bottom, there are function key definitions: F1=Help, F2=Split, F3=Exit, F5=Rfind, F6=Rchange, F7=Up, F8=Down, F9=Swap, F10=Left, F11=Right, F12=Cancel. The status bar at the bottom shows "12/080" and "Connected to remote server/host: 204.90.115.164 using lu/pool ICP00008 and port bc3".

Lámina 44 Roberto Gómez C.




## Tipos de data sets

---

- **Secuenciales (SDS)**
  - consiste de uno o más registros almacenados en orden físico y procesados en secuencia
- **Particionados (PDS)**
  - añade un nivel de organización a la estructura secuencial
  - colección de SDS llamados miembros
  - también contiene un directorio, que cuenta con entradas por cada miembro
  - PDSE: PDS extendido
- **VSAM**
  - Virtual Storage Access Method
  - aplica tanto al tipo de data set y el método de acceso
  - usado por las aplicaciones, no para programas fuente o módulos de ejecución

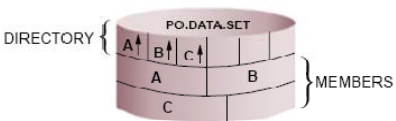
Lámina 45
Roberto Gómez C.




## Tipos de datasets no VSAM

---

### Particionados vs Secuenciales




**Partitioned organized**



**Physical sequential**


Lámina 46
Roberto Gómez C.



## Data Sets Secuenciales

- Simple estructura en sistema z/OS
- Uno o más registros que se encuentran en orden físico y son procesados secuencialmente.
- Nuevos registros son añadidos al final del data set.
- Ejemplos
  - un data set de salida para una impresora
  - un archivo de bitácoras
- Usuario define un data set secuencial con JCL con el valor de data seg organization de PS
  - DSORG = PS

Lámina 47 Roberto Gómez C.

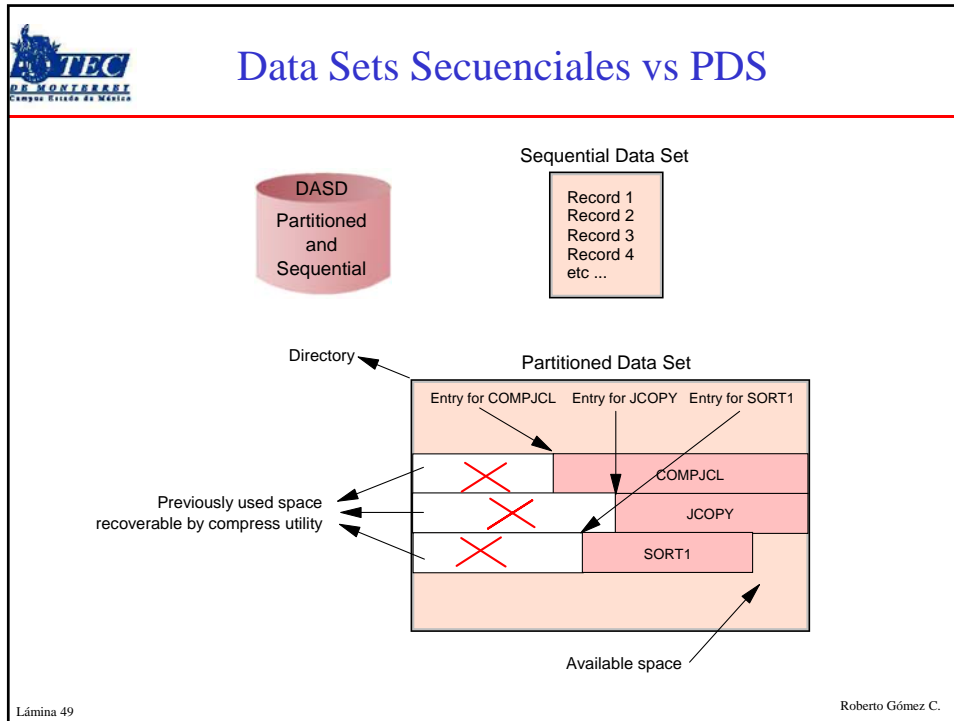


## Partitioned Data Sets


- Añade un nivel de organización al data set
- Compuesto por data sets secuenciales (*members*) y un directorio
  - cada miembro puede tener un nombre de máximo ocho caracteres
- Un PDS también se conoce como librería
  - guardar programas fuente, parámetros de control de aplicaciones y del sistema, JCL y módulos ejecutables
- Directorio cuenta con una entrada por cada miembro.
  - miembros listados alfabéticamente en el directorio pero pueden encontrarse en cualquier orden dentro de la librería

Lámina 48 Roberto Gómez C.






- 
- El directorio del PDS**
- Bloques contiguos de 256 bytes,
  - Al principio del Data Set
  - Cada bloque contiene:
    - 2 bytes de un campo contador
    - 3-21 entradas directorios
    - Una entrada directorio por cada miembro del PDS
      - 8 bytes nombre miembro
      - Posición inicial
      - Datos usuario (opcional)
  - Se puede contar con tantas entradas como puedan caber en 254 bytes (2 bytes) para el contador.
  - Longitud datos usuario determina cuantas entradas pueden caber.
  - Campo contador: numero de bytes usados (incluido el contador).
- Lámina 50 Roberto Gómez C.



## Ventajas PDS

- Agrupar data sets relacionados bajo un mismo nombre hace que el manejo de z/OS sea más fácil.
- Archivos almacenados como miembros pueden procesarse ya sea individualmente o como una unidad.
- Varios PDS pueden concatenarse para formar grandes librerías.
- Fáciles de crear con JCL o ISPF
- Fáciles de manipular con utilerías ISPF o comandos TSO.


Lámina 51 Roberto Gómez C.



## Desventajas PDS

- Espacio desperdiciado.
  - Espacio borrado no es reasignado
  - Reemplazo no usa el mismo espacio
- Tamaño directorio limitado
  - cuando se llena el directorio es necesario un nuevo PDS
- Búsquedas directorio lentas
  - búsquedas hechas en orden alfabético, mayor número de entradas, mayor tiempo de búsqueda
  - gran actividad I/O cada vez que un miembro es añadido

Lámina 52 Roberto Gómez C.




## Los PDSE

---

- Es un PDS extendido.
- Consiste de un directorio y cero o más miembros.
- Puede ser creado con JCL, TSO/E y ISPF.
- Data sets son almacenados en DASD no en cintas.
- Cada miembro puede contener hasta 15,778,639 registros.
- Puede contar con máximo 123 *extents*, pero no más allá de un volumen.
- Cuando se esta usando un directorio PDSE, este se deja en espacio de procesador para un acceso rápido.

Lámina 53 Roberto Gómez C.




## Características PDSE

---

- Directorio puede expandirse rápidamente conforme se necesite
  - limite de 522,236 miembros
- Cuenta con una estructura de directorio indexada
  - Cada entrada en el directorio apunta a la que sigue.
  - Las entradas no necesitan estar contiguas.
  - Pueden estar intercaladas con los bloques de datos de los miembros.
  - Búsqueda rápida para nombres de miembros
- Espacio borrado o movido es automáticamente reusado.


Lámina 54 Roberto Gómez C.



## PDS vs PDSE

- PDS data sets
  - forma simple y eficiente de organizar grupos relacionados de archivos secuenciales
- PDSE data sets
  - Similar a PDS, pero las ventajas incluyen
    - Espacio reclamado automáticamente cuando un miembro es borrado
    - Tamaño flexible del directorio
    - Puede compartirse
    - Búsqueda e inserción en directorio más rápida
    - Creación de múltiples miembros al mismo tiempo.
      - posible abrir dos DCBs al mismo PDSE y escribir dos miembros al mismo tiempo.


Lámina 55 Roberto Gómez C.



## ¿Y si ya no hay espacio para el data set?

- Cuando se asigna un data set se reserva una cantidad de espacio para este
  - Unidades de blocks, tracks, o cilindros en un disco de almacenamiento
- Si se usa mas del espacio reservado
  - mensaje: SYSTEM ABEND '0D37' o posiblemente B37 o E37.
  - si se encuentra en una sesión de edición, no se podrá salir de esta hasta que el problema se solucione
- Soluciones
  - si es un PDS, comprimirlo
  - asignar otro PDS más grande y copiarlo en él


Lámina 56 Roberto Gómez C.



## VSAM

- Virtual Storage Access Method
  - se refiere tanto al tipo de data set como al metodo de acceso
- Proporciona funciones más complejas que otros métodos de acceso a disco.
- Los registros de los discos son almacenados en un formato que no es entendible por otros métodos de acceso.
- Usado para aplicaciones
  - no para programas fuente, no JCL, ni módulos ejecutables
  - no pueden ser editados con rutinas, ni con ISPF

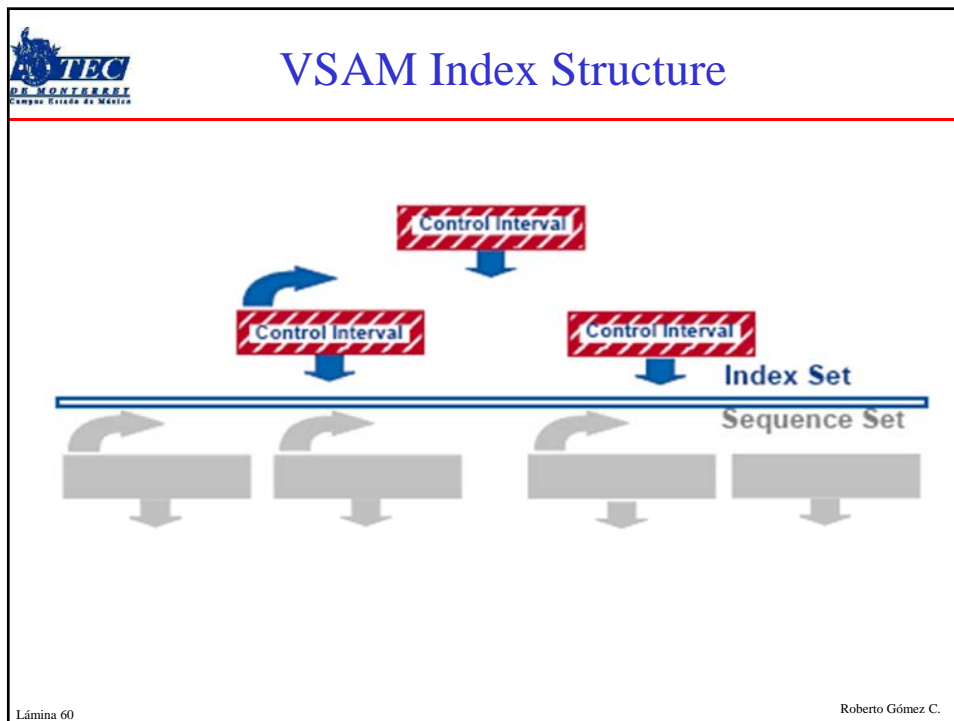
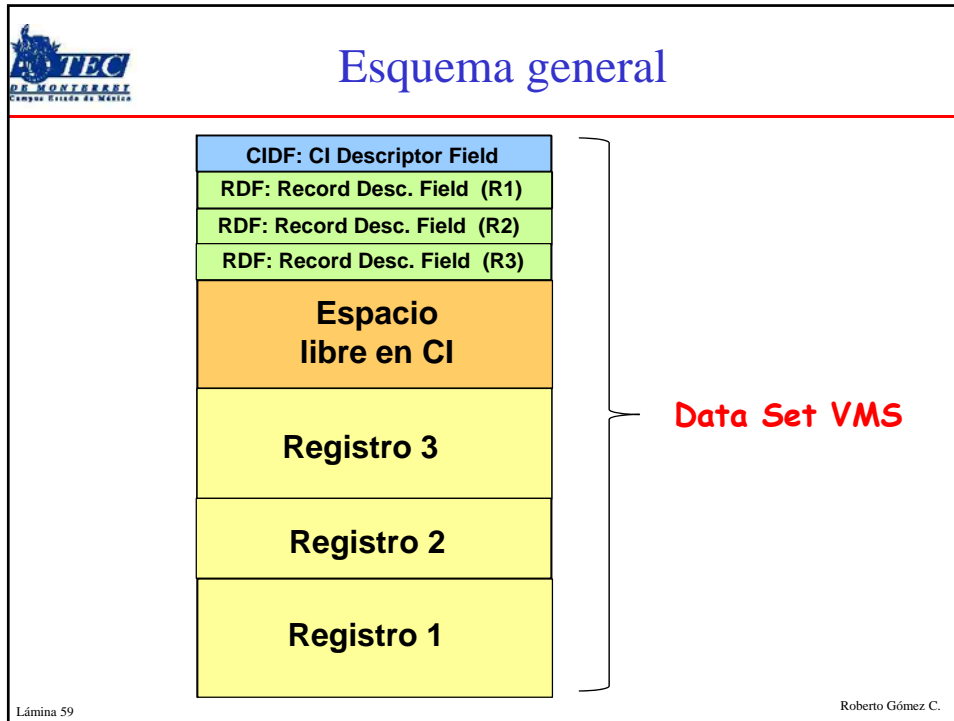
Lámina 57 Roberto Gómez C.



## Características VSM

- Un data set VSM Consiste de Control Intervals (CI), y Control Areas (CA).
- Tamaño CI y CA determinado por método acceso.
- CI contiene múltiples registros de datos, espacio sin usar, RDFs (Record Descriptor Field) y un CIDF (Control Interface Descriptor Field).
  - Los registros de datos son almacenados a partir de las direcciones bajas del Data Set.
  - Los RFDIS son almacenados en las direcciones altas, describe la longitud del registro asociado.
  - En medio de los dos hay espacio libre.
- Varios CI son colocados en una CA.

Lámina 58 Roberto Gómez C.






## Tipos organización registros

---

- VSAM puede organizar registros en cuatro tipos de datas sets
  - Key Sequence Data Set (KSDS)
  - Entry Sequence Data Set (ESDS)
  - Relative Record Data Set (RRDS)
  - Linear Data Set (LDS)

Lámina 61
Roberto Gómez C.

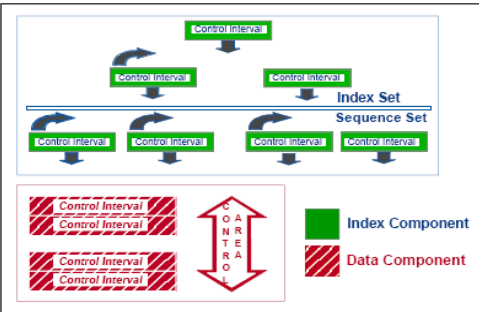


## VSAM Keyed Dataset

---

- Cada registro cuenta con una o más campos de llave
- Registro puede ser insertado o accedido a través del valor de la llave

**KSDS Structure** **Cluster**



Control Interval

Control Interval


Control Interval

SEARCH

Index Component

Data Component

Lámina 62
Roberto Gómez C.

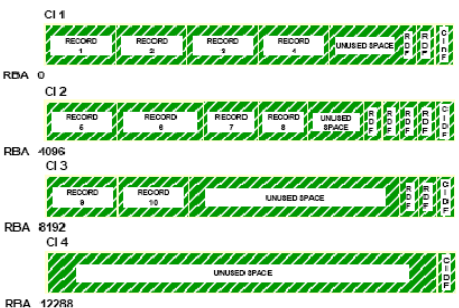


## VSAM Sequential Dataset = ESDS

---


- Registros en orden secuencial
- Registros accedidos secuencialmente
- Usados por IMS, DB2 y z/OS UNIX

**Entry Sequenced Data Set**



The diagram illustrates the structure of an Entry Sequenced Data Set (ESDS). It consists of four Control Intervals (CI 1 to CI 4). Each CI contains a sequence of records (RECORD 1 to RECORD 10) followed by unused space (UNUSED SPACE). The records are stored in a sequential order. The diagram also shows the Relative Byte Address (RBA) for each CI: CI 1 starts at RBA 0, CI 2 at RBA 4096, CI 3 at RBA 8192, and CI 4 at RBA 12288. The total length of the dataset is 12288 bytes.

Lámina 63
Roberto Gómez C.




## VSAM - RRDS

---

- Permite acceso de registros por número: registro 1, registro 2, etc
- Proporciona acceso aleatorio
- Asume que la aplicación tienen una forma de conocer los números de registro


**RELATIVE RECORD DATA SET (RRDS)**



The diagram illustrates the structure of a Relative Record Data Set (RRDS). It consists of four Control Intervals (CI 0 to CI 3). Each CI contains a grid of slots (SLOT 1 to SLOT 40). The slots are arranged in a grid pattern, allowing for random access to any record. The diagram also shows the Relative Byte Address (RBA) for each CI: CI 0 starts at RBA 0, CI 1 at RBA 4096, CI 2 at RBA 8192, and CI 3 at RBA 12288. The total length of the dataset is 12288 bytes.

Lámina 64
Roberto Gómez C.





## VSAM LDS

---

- Byte stream data set
- Único byte stream en z/OS y raramente usado en aplicaciones

LINEAR DATA SET (LDS)

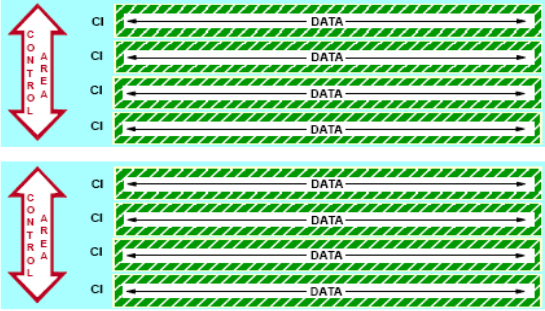



Lámina 65

Roberto Gómez C.



## Parámetros básicos para un VSAM dataset


---

DEFINE CLUSTER -

- (NAME (entryname))-
- CYLINDERS(primary secondary)|
- KILOBYTES(primary secondary)|
- MEGABYTES(primary secondary)|
- RECORDS(primary secondary) |
- TRACKS(primary secondary) -
- VOLUMES(volser[volser...])-
- DATA (parameters) -
- INDEX (parameters) -
- CATALOG (subparameters)

Lámina 66

Roberto Gómez C.




## Como son almacenados los datos en el sistema z/OS

---

- Datos son almacenados en un DASD, volumen de cinta magnética, o medio óptico.
- Es posible almacenar y retirar registros ya sea directa o secuencialmente.
- Es posible usar volúmenes DASD para almacenar datos y programas de ejecución, incluyendo el sistema operativo mismo, y por almacenamiento temporal de trabajo.
- Es posible usar un volumen para diferentes data sets, y reasignar o reusar espacio en el volumen.

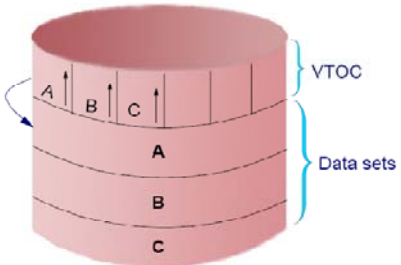
Lámina 67
Roberto Gómez C.



## Catalogos y VTOCs


---

- z/OS utiliza un catálogo y una VTOC (Volume Table Of Contents) en cada volumen DASD para manejar almacenamiento y asignación de data sets.
- VTOC
  - Lista los data sets en un volumen
  - Lista el espacio libre en el volumen.



The diagram illustrates a DASD volume. At the top, a table labeled 'VTOC' has three columns labeled 'A', 'B', and 'C'. Below the VTOC, three data sets labeled 'A', 'B', and 'C' are shown as stacked horizontal cylinders. A blue bracket on the right side groups the VTOC and the data sets together, with the label 'Data sets' pointing to the data sets.

Lámina 68
Roberto Gómez C.



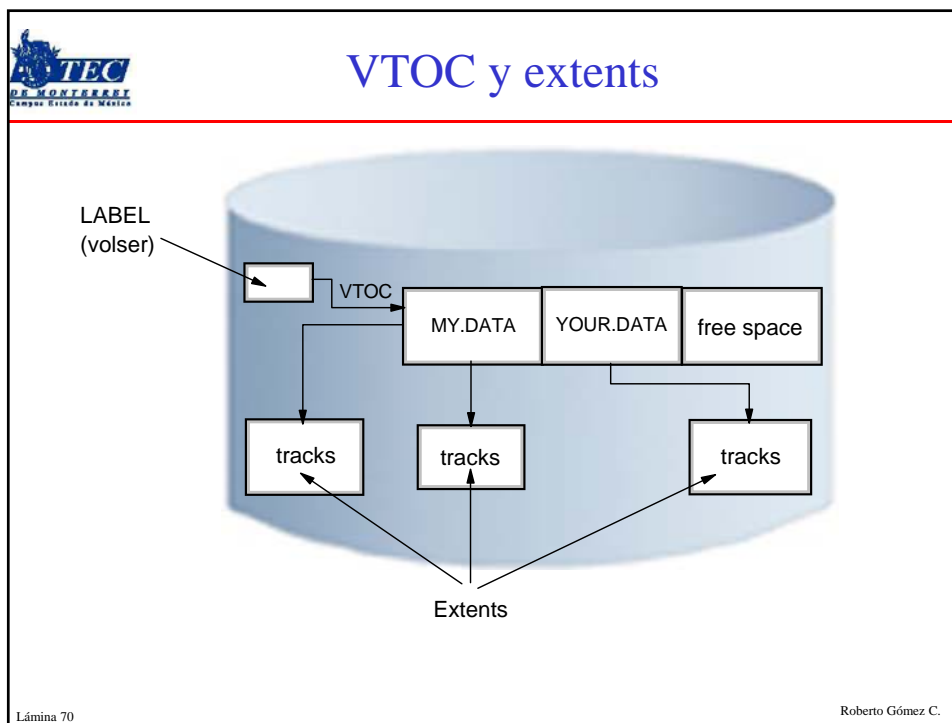
## Características VTOC

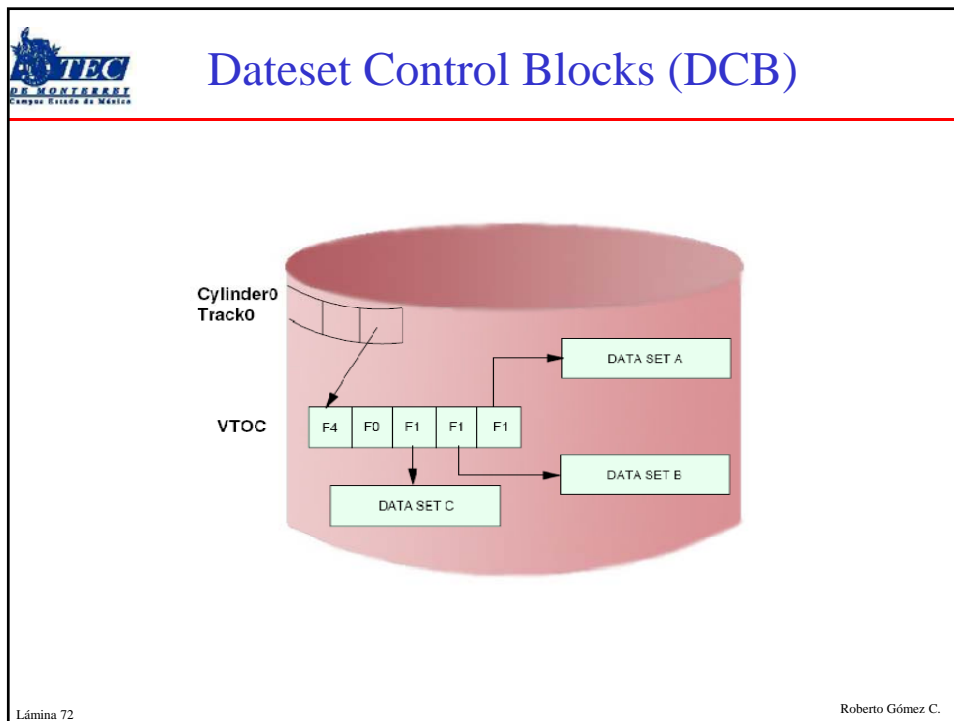
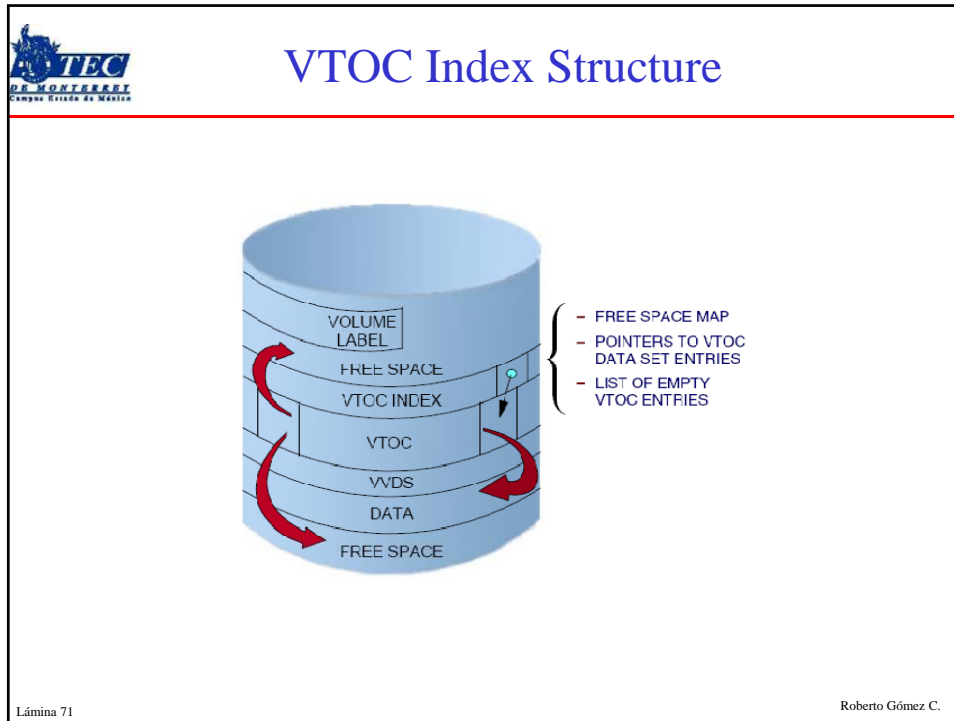
---


- Cuando volumen se inicializa con ICKDSF, se puede especificar la ubicación y tamaño de VTOC.
  - Tamaño puede ser variable de unos tracks a 100 tracks.
  - Más espacio en el volumen requiere mas espacio en el VTOC.
- Cuenta con entradas de todo el espacio libre en el volumen.
  - Bitmaps de espacio libre en el volúmen
- Posible crear un VTOC con un index
  - data set con nombre SYS1.VTOCIX.volser
  - entradas ordenadas alfabeticamente con el data set name apuntando a las entradas VTOC

Lámina 69

Roberto Gómez C.








## Los catalogos

---

- Un catalogo asocia un data set con el volumen en el cual el data set es asignado.
- Localizar un data set requiere
  - Nombre del data set
  - Nombre del volumen
  - Unidad (volumen device type)
- Cuando un data set se encuentra catalogado, este puede ser referenciado sin la necesidad de que el usuario especifique donde se encuentra almacenado el data set.
- Los data set pueden ser catalogados, descatalogados o recatalogados

Lámina 73
Roberto Gómez C.




## Datasets catalogados y no catalogados

---

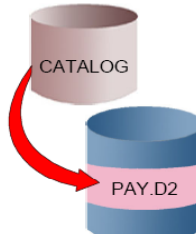
Uncataloged reference

```
// DD DSN=PAY.D1
      DISP=OLD
      UNIT=3390
      VOL=SER=MYVOL
```




Cataloged reference

```
// DD DSN=PAY.D2
      DISP=OLD
```



Notar el ‘//’ y los enunciados parm usados por JCL

Lámina 74
Roberto Gómez C.




## Catálogos maestros y de usuarios

---

- Sistema z/OS siempre tiene un catalogo maestro.
  - si solo es uno, este es el maestro y contiene las entradas de todos los data sets almacenados.
- No muy eficiente contar con un solo catalogo.
- Típicamente se cuenta con un catalogo maestro y numerosos catálogos de usuario.
- Catalogo usuario almacena el nombre y ubicación del data set (dsn/volume/unit)
  - el catalogo maestro usualmente almacena solo un data set HLQ con el nombre del catalogo del usuario que contiene todos los data sets con nombre precedidos con el HLQ.
- El HLQ se conoce se conoce como alias.

Lámina 75
Roberto Gómez C.



## Localizando un data set en MVS

---

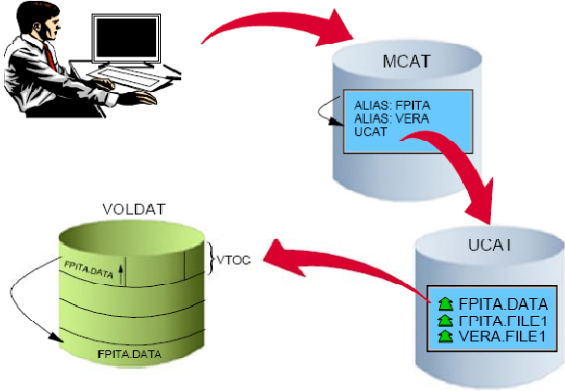
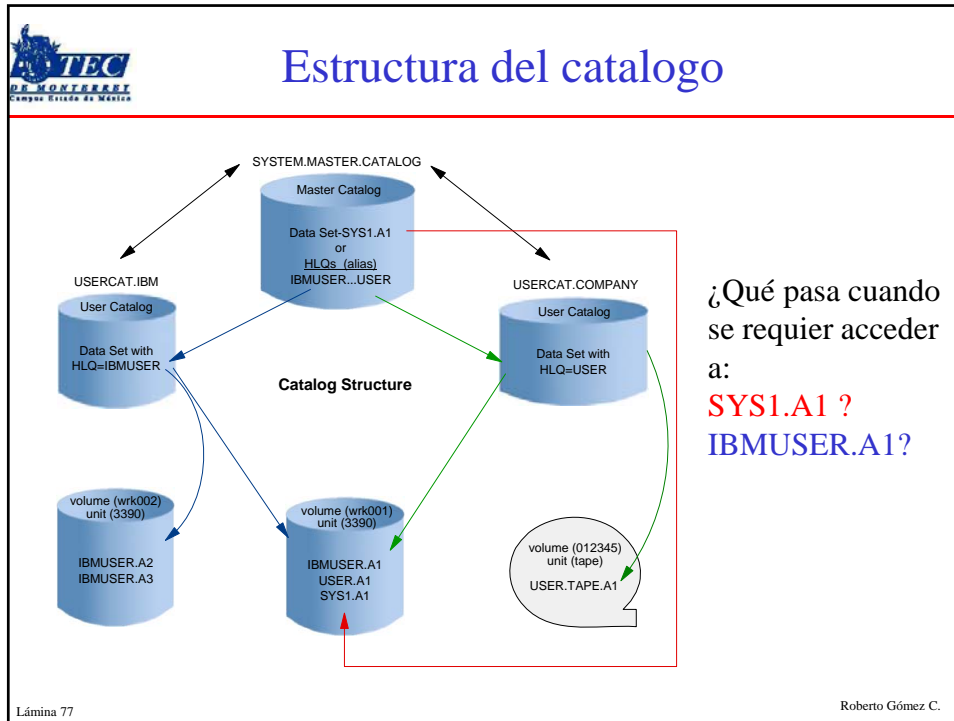



Lámina 76
Roberto Gómez C.



- 
- ¿Qué pasa si se pierde el catálogo?**
- La situación puede provocar un serio problema.
  - Programadores sistema definen un backup para el catálogo maestro.
  - Este catálogo maestro alterno se especifica durante el arranque del sistema.
  - Se recomienda que el programador de sistemas guarde el catalogo alterno en un volumen separado del volumen del catálogo maestro.
    - Protección en caso de que el volumen se estropee.
- Lámina 78 Roberto Gómez C.




## Generation Data Group

---

- Catalogar actualizaciones sucesivas o relacionadas
- Cada data set dentro de un GDG es llamado una generación o generation data set (GDS)
- Ventajas
  - todos por ser referenciados por un mismo nombre
  - sistema operativo los almacena en orden cronológico
  - generaciones no fechadas u obsoletas pueden ser eliminadas automáticamente por el sistema.


Lámina 79 Roberto Gómez C.



## Large Volume (own device type)

---

- Un “volumen largo” es más grande que un 3390-9
- El volumen más grande posible tiene 32,760 (3390) cilindros
- Sería un 3390-27 si cuenta con su propio tipo de dispositivo
  - Casi 28 GB




32760 Cyl

3390-27

Lámina 80 Roberto Gómez C.






## Manejo de datos en z/OS

---

- Manejo de datos involucra todas las siguientes tareas
  - asignación, monitoreo, migración, respaldo, recuperación y borrado
- Manejo de almacenamiento es hecho ya sea manualmente o a través de procesos automáticos (también es posible una combinación de ambos).
- En z/Os, DFSMS es usado para automatizar manejo de almacenamiento de datasets.

Lámina 81
Roberto Gómez C.



## DFSMS: Data Facility Subsystem Managed Storage

---

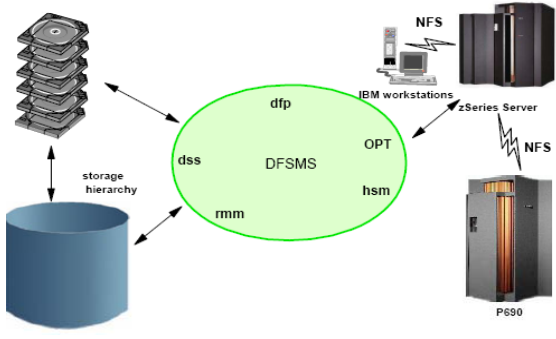




Lámina 82
Roberto Gómez C.



## El SMS (Storage Management Subsystem)

- Corazón del DFSMS
- Programador sistema define políticas que automatizan el manejo del almacenamiento y dispositivos de hardware
- Políticas definen características de ubicación de datos, objetivos, desempeño y disponibilidad
- El ISMF (Interactive Storage Management) proporciona interfaz para definir y mantener políticas.

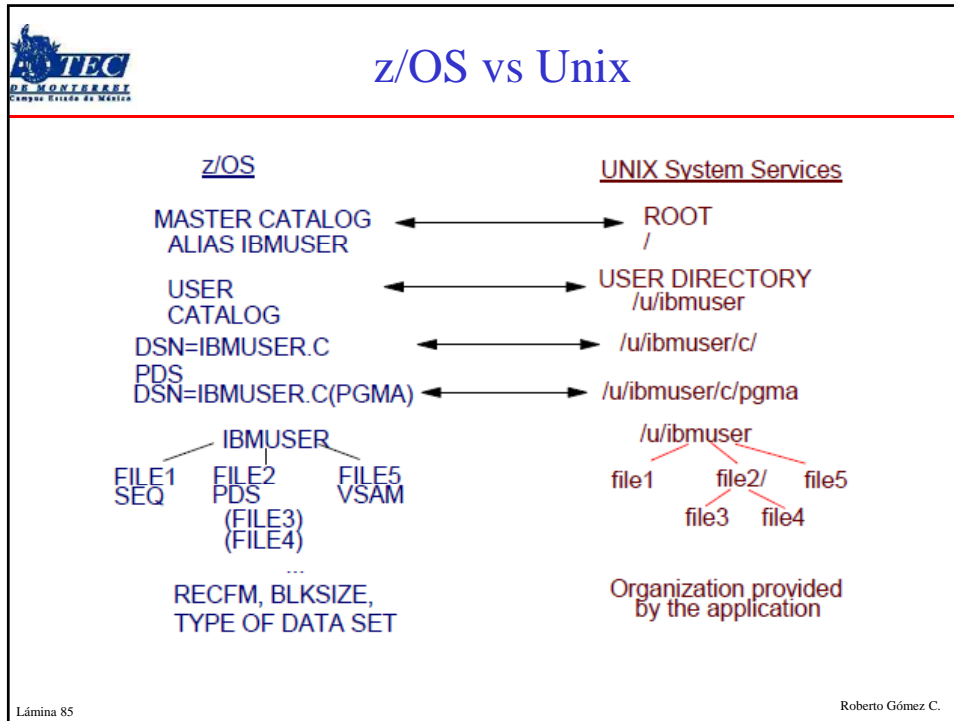
Lámina 83 Roberto Gómez C.



## z/OS UNIX file systems

- El servicio UNIX de z/OS (z/OS UNIX) permite a z/OS acceder a archivos UNIX
- Sistema archivos del UNIX de z/OS es jerárquico y orientado a bytes.
- Archivos en el sistema de archivos UNIX son archivos secuenciales que son accedidos como streams de byte
- Archivos UNIX y datasets pueden residir en el mismo volumen DASD.

Lámina 84 Roberto Gómez C.



**Manejo de Datasets**

Roberto Gómez Cárdenas

<http://homepage.cem.itesm.mx/rogomez>

rogomez@itesm.mx

Lámina 86 Roberto Gómez C.