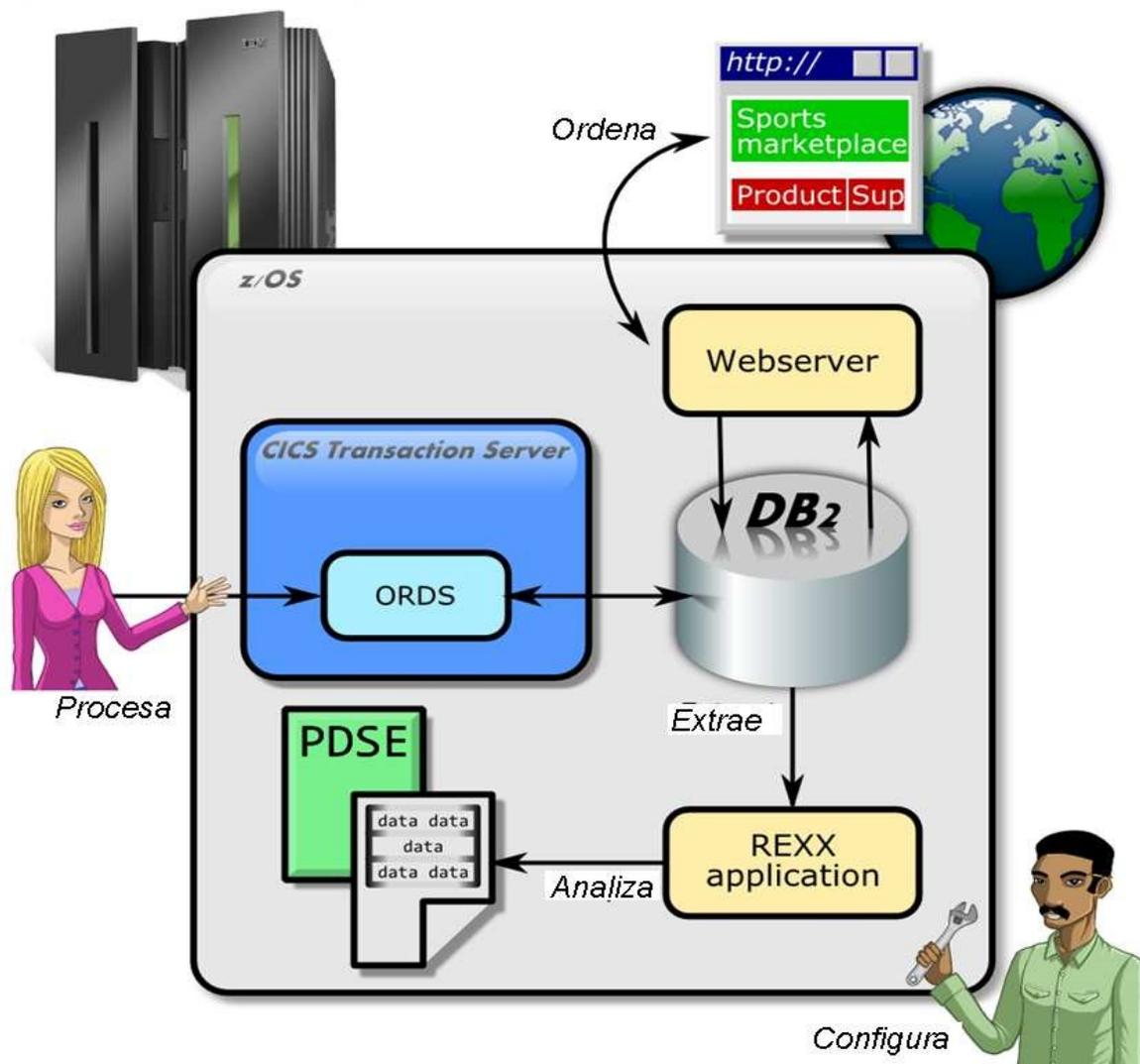


# Desafío Mainframe IBM para Estudiantes en Sudamerica

## Parte Dos

Tiempo estimado – de seis a ocho horas

La empresa en la que trabajas actúa como agente de varios proveedores y ofrece un sitio Web que los clientes pueden usar para adquirir los productos que ellos venden. En esta parte del Desafío Mainframe, usaremos el Gestor Transaccional CICS para actualizar información sobre pedidos, crearemos un trabajo batch para extraer datos de ventas de una base de datos DB2 y luego generaremos un informe de ventas a partir de él. Antes de todo esto, sin embargo, veamos brevemente algunas configuraciones de sistema para presentarte la manera en que un sistema z/OS puede ser configurado por sus administradores.



## Ayuda

Las siguientes referencias pueden resultar útiles para completar las tareas:

- Centro de Información de z/OS v1.11:  
<http://publib.boulder.ibm.com/infocenter/zos/v1r11/index.jsp>
- Centro de Información CICS Transaction Server v3.2:  
<http://publib.boulder.ibm.com/infocenter/cicsts/v3r2/index.jsp>

## COMENZANDO



Para probar tus nuevas habilidades en z/OS, me gustaría que trabajes con el equipo de procesamiento de pedidos pero primero Gemma te llevará con Carlos, uno de nuestros programadores de sistemas. Para verificar tu progreso y comprensión, tengo algunas preguntas para que respondas sobre la marcha.

Antes de empezar con esta parte de la prueba tienes que conseguir las preguntas que debes responder durante la marcha.

Las preguntas están a tu disposición en una hoja de preguntas y respuestas en el data set secuencial llamado ZOS.CONTEST1.PART2.ANSWER.SHEET.

1. **Usando ISPF, elabora una copia de este data set**, el que podrás completar a medida que avances en cada una de las tareas.

Para copiar un data set, usa la utilidad 3.3 del panel primario de ISPF. (Escribe =3.3 en la línea de comandos de ISPF.)

También puedes copiar miembros PDSE desde este panel poniendo el nombre del miembro entre paréntesis después del nombre del data set.



Nombra a tu data set secuencial **SSAxxxx.SSAxxxx.ANSWERS**  
 Debe estar asignado en pistas (**TRKS**)  
 Debe tener una cantidad primaria (primary quantity) igual a **1** y una cantidad secundaria (secondary quantity) igual a **1**  
 Debe tener 0 bloques de directorio (directory blocks)  
 El formato de registro debe ser **FB** (bloque fijo) y la longitud del registro debe ser igual a **80**  
 El tamaño del bloque debe ser igual a **32000**  
 El tipo de nombre del data set debe estar en blanco

1. **Asegúrate de que los primeros DOS calificadores de tu data set sean tu identificación de usuario**, de otro modo tus respuestas pueden no estar marcadas correctamente.

¡Si todo sale mal, no te preocupes! Borra el data set buscándolo primero en la utilidad 3.4 y luego escribiendo **D** en la columna de comando de la izquierda.  
¡Entonces intenta nuevamente!



Se te indicará cuando debes poder responder cada una de las preguntas, de todos modos lee los comentarios que se encuentran en la parte superior de la hoja de respuestas con cuidado antes de continuar.

1. **Ingresa tu identificación de usuario de z/OS** (es decir, SSAxxxx) en el espacio sobre la pregunta 1.

# Introducción a la Programación de Sistemas

Los sistemas mainframe están diseñados para que sean utilizados por una gran cantidad de personas. La mayoría de los que interactúan con mainframes son usuarios finales, personas que usan las aplicaciones albergadas por el sistema.

Sin embargo, a causa de la complejidad del software de mainframe que soporta a ese gran número de usuarios y aplicaciones, se necesita una gama de funciones para operar el mainframe en sí mismo.

Una de esas funciones es el de Programador de Sistemas. ¡Ese soy yo! Soy Carlos, el responsable de la instalación, personalización y mantenimiento de z/OS en nuestro mainframe.



z/OS ofrece muchos comandos y soluciones para ayudar al Programador de Sistemas a que haga esto. Ahora echaremos un vistazo a algunos de esos comandos y soluciones.

Cuando respondas las siguientes preguntas, podrás hallar de utilidad el libro MVS en el Centro de Información z/OS para comprender la sintaxis de un comando, para qué se usa y el resultado que produce.

Puedes encontrar el Centro de Información en:

<http://publib.boulder.ibm.com/infocenter/zos/v1r11/index.jsp>

# Usando SDSF

En esta sección podrás tener que moverte repetidamente entre dos aplicaciones diferentes, por ejemplo, entre el data set de la hoja de respuestas en el editor y SDSF para encontrar las respuestas a las preguntas. Puede ser frustrante el hecho de ir y venir entre diferentes aplicaciones, es por eso que hay una mejor manera de hacerlo:

Al presionar **F2** la pantalla se "dividirá" a la altura de tu cursor, lo que mostrará las dos pantallas separadas por una línea de puntos horizontales. Entonces podrás presionar **F9** para pasar de una pantalla a la otra.

Para maximizar el tamaño de pantalla disponible, coloca el cursor en la línea superior de la pantalla antes de presionar **F2**.

¡Nunca entenderás cómo viviste sin esto!



## Datos básicos sobre SDSF

*System Display and Search Facility* (SDSF) te brinda información que puedes usar para supervisar, administrar y controlar un sistema z/OS.

**Puedes usar SDSF para ver el registro del sistema y el estado de los trabajos que se están ejecutando en el sistema o que han terminado de ejecutarse, lo que incluye cualquier salida retenida de trabajos que aún no ha sido purgada (descartada).**

Se accede a SDSF desde el panel primario de ISPF con el comando **S**.

Desde allí, puede usarse **DA** para ver los trabajos activos (en ejecución) en el sistema, lo que incluye cualquier tarea iniciada, que es un tipo especial de trabajo.

**ST** puede usarse para ver el estado de los trabajos, los que están activos, los que aún no fueron procesados y los que están terminados.

Una vez que estés viendo una lista de trabajos, en la indicación de ingreso de comandos puedes filtrar la lista de trabajos para ver solamente los tuyos. Para eso, ingresa:

**OWNER SSAxxxx**

Para revertir el filtro y volver a ver todos los trabajos, ingresa:

**OWNER \***

De manera similar, puedes filtrar la lista de trabajos por nombre. Para eso, ingresa:

**PREFIX ABC\***

con lo que solo aparecerán aquellos trabajos cuyos nombres comiencen con las letras ABC..

Para quitar este filtro, ingresa:

**PREFIX \***

Puede usar ambos filtros, el de propietario de trabajos y el de prefijo al mismo tiempo.

Ellos son de particular utilidad ya que ocultan trabajos en el sistema en los cuales no estás interesado o no tienes autorización para verlos.

No olvides que SDSF recuerda los filtros que no estás usando. Si no ves un trabajo que esperabas ver, pon en ambos filtros un solo asterisco (\*) para que aparezcan todos los trabajos.

En los paneles DA y ST se te presentará una lista de trabajos. Usa las teclas **PF7** y **PF8** para subir y bajar por la página.

Para ver más información acerca de un trabajo en particular, puede ingresar **S** en la columna de la izquierda (NP) y presionar **Ctrl**.

Esto puede suministrarte una gran cantidad de información y así, alternativamente, puedes ingresar un signo de interrogación (?), lo que te proporcionará un listado de las diversas secciones de la salida. Puedes usar **S** para seleccionar esas secciones individualmente.

Usa la tecla **PF3** para salir de la pantalla en la que te encuentras y regresar a la pantalla previa.

Los filtros de prefijo y de propietario no aparecen por defecto en ninguna parte, por lo que si dejas activados los filtros es probable que termines con una lista de trabajos en blanco. ISPF recuerda los filtros que hayas usado entre sesiones; busca en el menú Options (Opciones) en la parte superior de la pantalla y selecciona la opción 5 para verlos.



```

Display Filter View Print Options Help
-----
HQX7708 ----- SDSF PRIMARY OPTION MENU -----
COMMAND INPUT ==> st_ SCROLL ==> CSR

DA Active users INIT Initiators
I Input queue
O Output queue ULOG User session log
H Held output queue
ST Status of jobs

LOG System log

END Exit SDSF

Licensed Materials - Property of IBM

5694-A01 (C) Copyright IBM Corp. 1981, 2003. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

```

Display Filter View Print Options Help
-----
SDSF STATUS DISPLAY ALL CLASSES LINE 49-64 (2441)
COMMAND INPUT ==> owner UKxxxxx_ SCROLL ==> CSR
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
NP JOBNAME JobID Owner Prty Queue C Pos SAff ASys Status
BPXAS STC00679 OMVS 15 EXECUTION MVS1 MVS1
BPXAS STC00680 OMVS 15 EXECUTION MVS1 MVS1
BPXAS STC00681 OMVS 15 EXECUTION MVS1 MVS1
IBMSM STC00682 STC 15 EXECUTION MVS1 MVS1
CPSMCAS STC00683 SYSSTC 15 EXECUTION MVS1 MVS1
RMFGAT STC00684 STC 15 EXECUTION MVS1 MVS1
CICSAOR3 STC00685 STC 15 EXECUTION MVS1 MVS1
DSNBMSTR STC00686 SYSSTC 15 EXECUTION MVS1 MVS1
DSNBIRLM STC00688 SYSSTC 15 EXECUTION MVS1 MVS1
DSNBDBM1 STC00689 SYSSTC 15 EXECUTION MVS1 MVS1
OSNMPPD STC00694 OSNMPPD 15 EXECUTION MVS1 MVS1
SMTP STC00693 SMTP 15 EXECUTION MVS1 MVS1
BPXAS STC00695 OMVS 15 EXECUTION MVS1 MVS1
IMSASCI STC00697 SYSSTC 15 EXECUTION MVS1 MVS1
DSNBDIST STC00698 SYSSTC 15 EXECUTION MVS1 MVS1
IMSARM STC00700 SYSSTC 15 EXECUTION MVS1 MVS1

F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

# Introducción a IPL

Cuando enciendes tu computadora de escritorio o portátil en casa, el sistema operativo se inicia por medio de un proceso conocido como "booting" ó arranque. Este proceso usa un registro de arranque en disco para localizar al sistema operativo e iniciar el proceso de inicio. En z/OS esto se conoce como "IPL" (Initial Program Load) y el equivalente de un registro de arranque se llama "load member" (miembro de carga).

IPL son las iniciales de "Initial Program Load" (Carga del Programa Inicial) pero las personas lo usan como un verbo: "dar IPL" ó "IPLear" significa "iniciar el mainframe". Es un buen punto para empezar a aprender sobre programación de sistemas, por eso echaremos un vistazo a la información suministrada por nuestro sistema z/OS cuando damos IPL.



Desde el z/OS Primary Option Menu (Menú de Opción Primario de z/OS) en ISPF selecciona la opción para SDSF y luego la opción ULOG que abre el Registro de Sesión del Usuario, el que puede utilizarse para ver justo en la salida de los comandos que emitas. Esto evita tener que subir y bajar por el registro principal del sistema, el que puede tener muchísimas entradas, para localizar las líneas correspondientes.

Para visualizar la información de IPL para el sistema z/OS system emite el comando **/DISPLAY IPLINFO** o, si lo prefieres, **/D IPLINFO** en su lugar. La respuesta será dirigida a tu registro de sesión; tómate un momento y léelo.

## 1. Ahora responde la pregunta 1.

```
Display Filter View Print Options Help
-----
SDSF ULOG CONSOLE UKxxxxx LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> /display iplinfo_ SCROLL ==> CSR
***** TOP OF DATA *****
TESTMVS 2010284 xx:xx:xx.xx ISF031I CONSOLE UKxxxxx ACTIVATED
***** BOTTOM OF DATA *****

F1=HELP F2=SPLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

MA a 04/037
```

Además del miembro de carga, el sistema también requiere uno o más data sets conocidos colectivamente como Librería de Parámetros del Sistema (PARMLIB). Estos data sets contienen varios miembros que detallan información adicional sobre configuración para los diferentes aspectos del sistema.

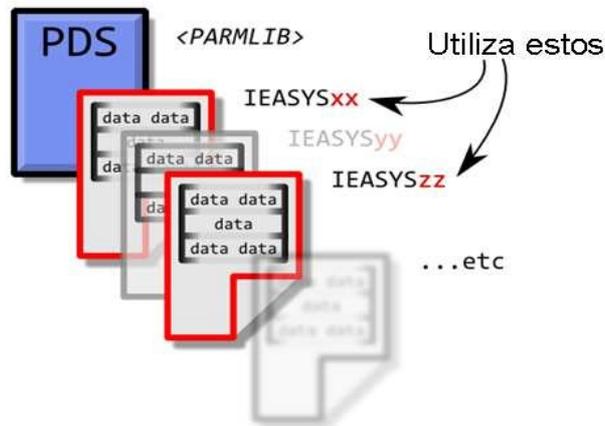
Si estás más familiarizado con Linux o UNIX, PARMLIB en z/OS es parecido al directorio /etc/.



**1. Ahora responde la pregunta 2.**

Puedes ver una lista de los comandos de visualización bajo el título *MVS – System Commands (MVS – Comandos del sistema)* en el Centro de Información z/OS.

Los parámetros de configuración PARMLIB en z/OS están divididos en diferentes miembros para diferentes aspectos del sistema. z/OS construye los nombres de estos miembros por medio de prefijos específicos combinados con sufijos personalizados. La mayoría de estos sufijos están definidos en la lista de parámetros del sistema, la que puede hallarse en uno o más miembros cuyos nombres comienzan con IEASYS.



El uso de sufijos como este permite que el operador haga IPL a un sistema z/OS con diferentes parámetros simplemente especificando distintos sufijos para los miembros IEASYS. Los sufijos IEASYS predeterminados por lo general están especificados en el miembro de carga pero un operador puede ignorar esto cuando se inicia un IPL.

El comando **DISPLAY IPLINFO** que emitiste hace un minuto desplegó una lista de los sufijos usados por este sistema. Busca la salida en la parte superior del ULOG, deberás ver una línea para IEASYS LIST.



Deberás ver que esta lista contiene una entrada para el sufijo 00; el (OP) a continuación, de estar presente, indica que fue especificado por el operador, no por el miembro de carga. A partir de esto podemos determinar que cuando el sistema fue inicializado, leyó un miembro llamado IEASYS00.

### 1. Localiza y examina este miembro en el data set PARMLIB.

Deberás poder ver cómo están definidos los sufijos para los demás parámetros de configuración.

### 1. Ahora responde la pregunta 3.

## Recursos del Sistema

Al comprar una computadora personal, a muchos les gusta leer las especificaciones, tales como cuánta memoria tiene. z/OS se refiere a la memoria (RAM) como almacenamiento real o central. La máquina física System z puede ejecutar una gran cantidad de sistemas z/OS conocidos como particiones lógicas (LPAR). A cada LPAR se le ha asignado un subconjunto de recursos físicos, ya sea de manera exclusiva o de manera compartida con otras LPAR.

Para visualizar la cantidad de almacenamiento real que ha sido puesta a disposición de la LPAR para el sistema z/OS que estás utilizando, puedes usar el comando `DISPLAY M=STOR`.

### 1. Emite este comando y luego responde la pregunta 4.

z/OS también suministra un entorno UNIX conocido como Sistema de Servicios UNIX (USS) o OMVS (solía ser llamado OpenEdition MVS). Tendrás la oportunidad de usar USS en la parte 3 de la prueba.

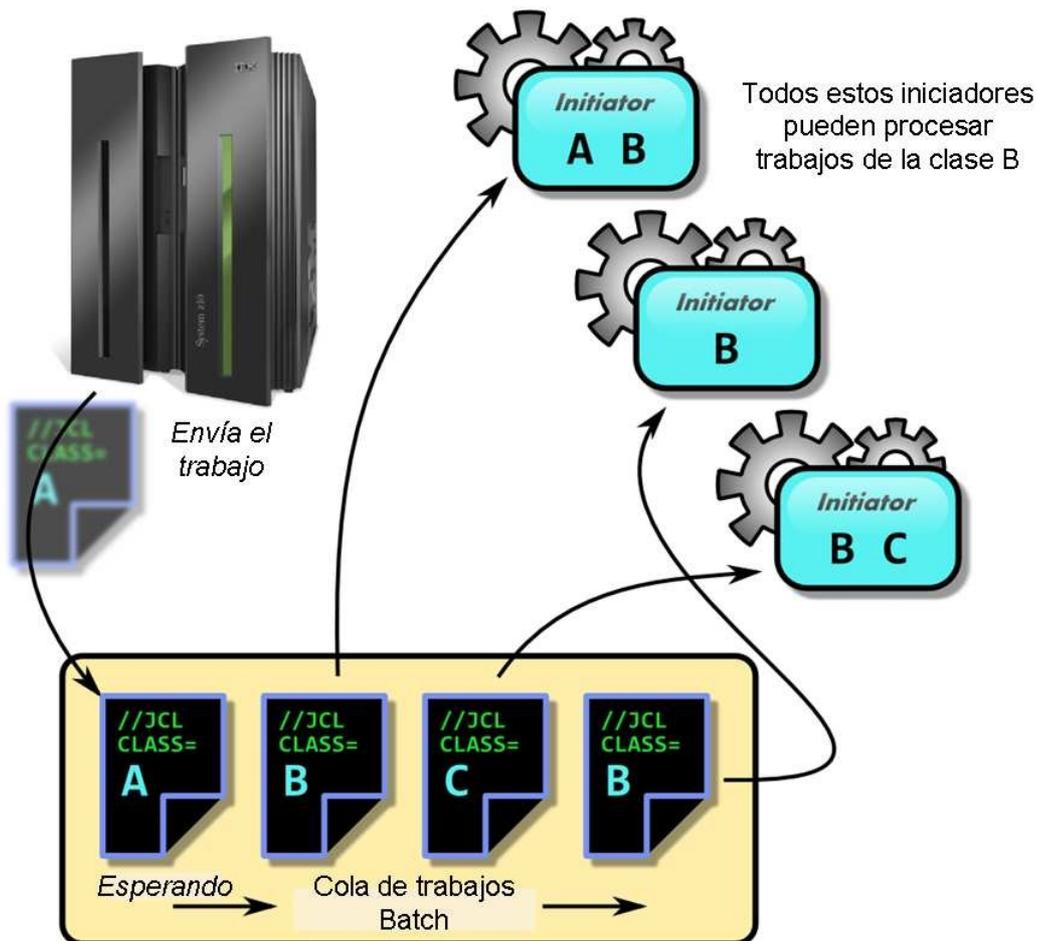
Al igual que en los sistemas UNIX, puedes configurar muchas opciones como por ejemplo cuántos procesos pueden ejecutarse simultáneamente, lo cual no está relacionado con el número de trabajos simultáneos que identificaste anteriormente. Esto puede ser encontrado en uno de los miembros de `PARMLIB` aunque es mucho más simple emitir el comando `DISPLAY OMVS,L` para ver los límites configurados.

### 1. Emite este comando y luego responde la pregunta 5.

# Trabajos Batch

En esta parte de la prueba usarás trabajos batch para hacer tareas. Para ejecutar un trabajo batch tienes que enviarlo y luego quedará en cola para su ejecución posterior hasta que un *iniciador* (un proceso que ejecuta trabajos) quede libre.

Para ayudar al equilibrio del sistema y dar prioridad a los diversos trabajos que podrían ser enviados, a cada trabajo se le asigna una *clasificación* (clase) y cada iniciador está configurado para ejecutar trabajos de una o más clases. Esto permite al Programador del Sistema configurar fácilmente el sistema para permitir, por ejemplo, que el doble de trabajos de una clase se ejecuten al mismo tiempo que los de otra clasificación de menor importancia.



Si se envía un trabajo para una clase para la cual no hay un iniciador, quedará en cola indefinidamente hasta que sea purgado o se ponga un iniciador a su disposición.

Puede ver los diversos iniciadores que han sido configurados en nuestro sistema z/OS usando otra opción del SDSF.  
¿Adivinas cuál es?



**1. Sal del registro de sesión de usuario y selecciona esta opción desde el menú SDSF Primary Option.**

Desde esta pantalla puedes ver los iniciadores, su estado y el tipo de clases para cuya ejecución fueron configurados. Un iniciador *activo* está ejecutando actualmente un trabajo, uno *inactivo* está actualmente detenido (esperando un trabajo) y uno *purgado* está actualmente deshabilitado.

**1. Ahora responde la pregunta 6.**

Hay mucho más en programación de sistemas que lo que has visto aquí, pero creo que por ahora es suficiente. Te dejo con Gemma de nuevo; ya puedes ejecutar algunos trabajos propios. ¡No dañes nada!



## Parte 2 - Tareas

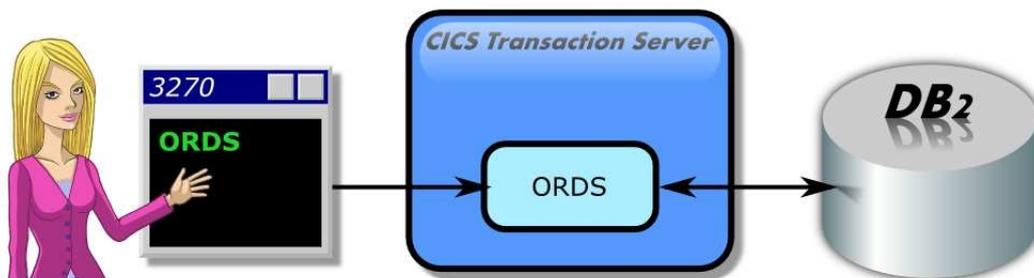
¡Bienvenido al equipo de procesamiento de pedidos!

...o, como me gusta llamarlo, ¡el increíble equipo de procesamiento!



La estrategia de nuestra empresa es ofrecer un sitio Web que permita a los clientes hacer pedidos de productos vendidos por otros revendedores. El equipo de procesamiento de pedidos puede recibir pedidos recientes y modificar su estado a medida que los procesa.

La información sobre los productos vendidos por cada revendedor y los pedidos realizados por los clientes está almacenada en una base de datos DB2. El equipo de procesamiento de pedidos usa una transacción CICS que se conecta con la DB2 para extraer y actualizar los datos necesarios.



# Introducción al CICS

El gestor transaccional *CICS* (Customer Information Control System) es un software de uso general de procesamiento de transacciones de IBM para z/OS. Es un poderoso servidor de aplicación que satisface las necesidades de procesamiento de transacciones de empresas grandes y pequeñas. También agrega soluciones de z/OS y System z para brindar alta disponibilidad y escalabilidad a bajo costo por transacción; da soporte a grandes volúmenes de transacción con un tiempo de respuesta rápido y consistente.

El procesamiento de transacciones es un estilo de computación, ejecutado por lo general por grandes servidores que dan soporte a aplicaciones interactivas. En el procesamiento de transacciones, el trabajo está dividido en operaciones individuales e indivisibles llamadas transacciones. Por ejemplo, extraer dinero de un cajero automático y actualizar el balance en una base de datos. Por el contrario, el procesamiento batch es un estilo de computación en el cual uno o más programas procesan una serie de registros (un lote *batch*) con muy poca o ninguna acción por parte del usuario u operador. Por ejemplo, la impresión de resúmenes bancarios para cada titular de cuenta y su envío posterior por correo.

Un *middleware* de procesamiento de transacciones tal como CICS permite que los programadores de aplicaciones se concentren en escribir un código que brinde soporte al negocio protegiendo los programas de aplicación de los detalles de administración de la transacción:

- Administra el procesamiento simultáneo de transacciones.
- Hace posible que se compartan los datos.
- Asegura la integridad de los datos.
- Administra el establecimiento de prioridades de ejecución de transacciones.

Cuando se empieza a procesar una transacción, CICS ejecuta un programa asociado con la transacción. Ese programa puede transferir control a otros programas en el curso de la transacción, lo que hace posible ensamblar aplicaciones modulares que consistan en muchos programas CICS.

En cualquier momento, en un sistema CICS, muchas instancias de una transacción pueden ejecutarse al mismo tiempo. Sería terrible tener que esperar a que otras personas terminen de pagar antes de poder comprar algo en línea.

## Datos básicos sobre CICS

Es un buen momento para brindarte asistencia sobre el uso de CICS.

Una transacción CICS se inicia presionando la tecla **CLEAR\*** para borrar la pantalla. Luego ingresa el identificador de la transacción, solo o seguido de datos, en la línea de comandos en la pantalla. La línea de comandos es una línea única, ubicada por lo general en la parte superior de la pantalla.

Puedes escribir únicamente el identificador de la transacción y seguir las indicaciones hasta estructurar un comando completo de transacción o puedes escribir el comando completo de transacción en la línea de comandos. Si no ingresas suficiente información o si la información ingresada es incorrecta, se te pedirá que completes o corrijas tu entrada.

Por ejemplo, en la siguiente transacción, **CESF** es el identificador de la transacción y **GOODNIGHT**, los datos adicionales.

**CESF GOODNIGHT**

Esta transacción se utiliza para cerrar sesión en CICS.

```
cesf goodnight_
```

Cuando la transición comience, procesará los datos adicionales. Al finalizar esta transacción, obtendrás el siguiente mensaje:

**STATUS: SESSION ENDED**

Una vez que la transacción haya finalizado, presiona la tecla **CLEAR**\* para que la pantalla quede en blanco y lista para la siguiente transacción.

Para finalizar una transacción, usa la tecla **PF3**.

\* La tecla especial **CLEAR** de los emuladores 3270 no existe en la mayoría de los teclados. Sin embargo, los emuladores 3270 pueden ofrecerla por medio de un menú contextual (clic derecho) o te permiten asignarla a una tecla de tu agrado, a menudo y por defecto a la tecla **Pause/Break**.

Puedes encontrar más información sobre CICS en el Centro de Información de [CICS Transaction Server v3.2](#).

## Uso de CICS

Abre una segunda sesión en 3270. En lugar de ingresar **TSO** en la indicación **SELECT APPLICATION**, ingresa **LOGON APPLID(CICSZ057)**.

Entonces, deberá aparecer la siguiente pantalla de inicio de sesión de CICS:

```
Signon to CICS                                APPLID CICSZ057

Welcome to the UK Mainframe Challenge CICS System for Part 2

Type your userid and password, then press ENTER:

  Userid . . . . UKxxxxxx   Groupid . . . . _____
  Password . . . .
  Language . . . . ___
  New Password . . . .

DFHCE3520 Please type your userid.
F3=Exit

Má a                                           10/033

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

Má a                                           22/014

Userid : UKxxxxxx
Groupid : xx:xx
Language : 3278
New Password : 1
Language : ENGLISH
Groupid : ISR
Logon : SYSUSER
Prefix : UKxxxxxx
ID : TESTMVS
System : UNIVER
Version : ISPF 5.6
```

1. **Ingresas tu identificación de usuario y presiona Enter** (recuerda que es la tecla **Ctrl**).

La pantalla deberá borrarse, salvo por el siguiente mensaje cerca de la parte inferior:

DFHCE3549 Sign-on is complete (Language ENU).

## Visualización y actualización de pedidos en CICS

1. **Una vez que tu sesión ha iniciado con éxito en CICS, ejecuta una transacción llamada ORDS.**

Esta es la transacción que usamos para mostrar pedidos recientes de los clientes. Cambiaremos el estado de un par de pedidos, luego tengo más preguntas para que respondas.



El artículo TN032 de Camping Stuff (Artículos de campamento) fue pedido por S Carter hace mucho tiempo y ahora ella decidió cancelar su pedido.

1. **Cambia el estado de este pedido, de On Order a Cancelled.**

El pedido para el artículo GF157 de Sally's Sports hecho por J Bloggs fue cancelado accidentalmente por uno de tus colegas.

1. **Cambia el estado de este pedido, de Cancelled a Packing.**

1. **Ahora responde las preguntas 7 a 9.**

# Introducción a JCL

¿Cómo realizarías una tarea que te hayan asignado? Yo la desglosaría en uno o más pasos de tal modo que cada uno de ellos complete una parte de un conjunto más grande. A veces necesitaré diferentes herramientas y recursos para cada paso.



El Lenguaje de Control de Trabajos "Job Control Language" (JCL) es la manera en la que describirás esta información a z/OS, según el cual un trabajo representa una tarea.

Para cada trabajo que envíes, tendrás que indicarle a z/OS en dónde buscar la entrada apropiada, cómo procesarla (es decir, qué programa o programas ejecutar) y qué hacer con la salida resultante. Puedes usar JCL para transmitir esta información a z/OS a través de un conjunto de instrucciones conocidas como instrucciones de control de trabajos.

El conjunto de instrucciones de control de trabajo de JCL es muy grande, lo que le permite suministrar una gran cantidad de información a z/OS. La mayoría de los trabajos, sin embargo, pueden ejecutarse usando un subconjunto muy pequeño de esas instrucciones de control.

En cada trabajo, las instrucciones de control están agrupadas en pasos de trabajo. Un paso de trabajo consiste en todas las instrucciones de control necesarias para ejecutar un programa.

Si un trabajo necesita ejecutar más de un programa, el trabajo contendrá un paso de trabajo diferente para cada uno de ellos.

## Instrucciones de control de trabajos

Cada trabajo debe contener:

- Una instrucción de trabajo **JOB** que marque el comienzo de un trabajo y le asigne un nombre. Esta instrucción también se usa para suministrar cierta información administrativa, lo cual incluye información sobre seguridad, contabilidad e identificación. Cada trabajo tiene una y solo una instrucción **JOB**.

- Por lo menos una instrucción **EXEC** (ejecutable) que indique el comienzo de un paso de trabajo, un nombre para ese paso de trabajo y el programa o procedimiento que se debe ejecutar. Tu puedes agregar diversos parámetros a la instrucción **EXEC** para personalizar la manera en que el programa se ejecuta y las condiciones bajo las cuales el paso debería ejecutarse.

La mayor parte de los programas requiere alguna entrada o genera alguna salida, por eso por lo general contienen:

- Una o más instrucciones **DD** (definición de datos) que identifique y describa los datos de entrada y de salida que se utilizarán en el paso. Esta instrucción puede ser usada para solicitar un data set previamente creado, para definir un nuevo data set, para definir un data set temporal o para definir y especificar las características de la salida

Cada instrucción de control de trabajo tiene cinco campos:

- 1.** Un campo de identificador, que por lo general consiste en dos barras diagonales (**//**). Toda la línea es tratada como un comentario si detrás de las dos barras hay un asterisco (**//\***).
- 2.** Un campo de nombre, que identifica a la instrucción de modo que se pueda hacer referencia a ella más tarde.
- 3.** Un campo de operación que identifica el tipo de instrucción, por ejemplo **JOB**, **EXEC**, **DD**.
- 4.** Un campo de parámetro.
- 5.** Lo que esté ubicado después del campo de parámetro será tratado como un comentario y como tal, será ignorado.

Las instrucciones JCL solo pueden codificarse hasta la columna 71. Puedes preguntarte por qué JCL tiene este formato: históricamente, se ingresaba con tarjetas perforadas que tenían 80 columnas, las últimas 8 de las cuales se usaban para números de línea.

El diseño de JCL puede parecer confuso cuando se ve por primera vez. Por suerte, el editor ISPF puede colorear la sintaxis de JCL si escribes **HILITE JCL** (o **HILITE AUTO**) al editar (no al examinar) un miembro de JCL.



Para más información sobre JCL, puedes consultar la sección [JCL Reference](#) del libro MVS en el Centro de Información de z/OS.

# Comenzando

Para la Parte 2 vamos a usar tanto JCL como REXX, por eso podemos crear data sets para almacenarlos. Por ahora, no pondremos nada en el data set REXX.  
¡Si ya olvidaste cómo hacerlo, vuelve a la Parte 1!



## 1. Usa ISPF para crear dos nuevos PDSE llamados **SSAxxxx.PARTTWO.JCL** y **SSAxxxx.PARTTWO.REXX**

Deben estar asignados en pistas (**TRKS**)

Deben tener una cantidad primaria igual a **1** y una cantidad secundaria igual a **1**

El formato de registro debe ser **FB** (fijo bloqueado) y la longitud del registro debe ser igual a **80**

El tamaño de bloque debe ser igual a **32000**

El tipo de nombre del data set debe ser **LIBRARY**

## Extracción de los Datos del pedido desde DB2



La gerente de Ventas y Comercialización solicita un informe a la fecha de ventas totales de este año para cada revendedor. También está interesada en los artículos de mayor y menor venta por revendedor para que se puedan dirigir campañas de comercialización.

Hicimos algo similar a esto el año pasado... Usamos una aplicación existente para extraer datos de ventas de la base de datos, luego ejecutamos otro programa para generar el informe a partir de los datos extraídos. Ambos fueron ejecutados en trabajos batch usando JCL. Hagámoslos funcionar de nuevo...



El programa existente para extraer todos los datos de ventas desde inicio del año de la base de datos DB2 se llama DB2EXTR y está almacenado en ZOS.CONTEST1.PART2.LOAD. Para ejecutarlo, tendrás

que enviar un trabajo JCL.

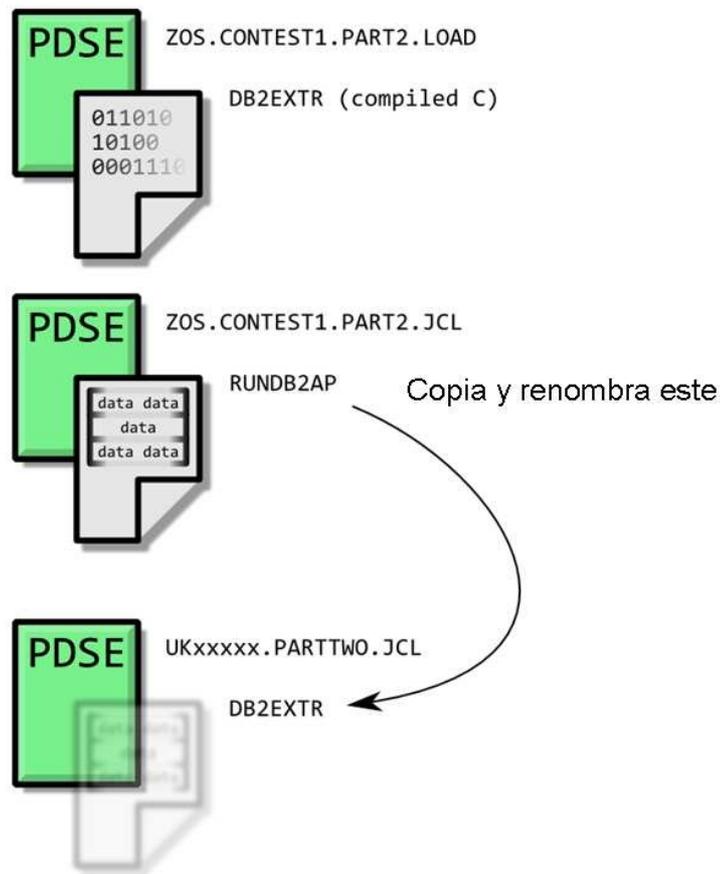
No hay ningún JCL para ejecutar el primer programa; tendrás que escribir algo tu mismo.

No te preocupes, casi todo el mundo escribe JCL desde cero en estos días. ¡Es mucho más fácil que modificar un trabajo que ya existe!



`ZOS.CONTEST1.PART2.JCL(RUNDB2AP)` es un ejemplo de un trabajo que ejecuta una aplicación DB2, por eso es un buen punto de partida para tu trabajo.

1. **Copia esto en tu librería JCL y cámbiale el nombre a DB2EXTR.**



Puedes cambiar el nombre del nuevo miembro copiado ya sea dándole un nombre diferente en el panel ISPF del data set copiado o poniendo `r` junto al miembro. Si eliges la segunda opción, deberás escribir `REFRESH` en la línea de comandos para actualizar la lista con el nuevo nombre del miembro.

```

Menu RefList Utilities Help
-----
COPY      From ZOS.UK.CHAL2010.JCL (RUNDB2AP)
Command ==> _____
More: +

Specify "To" Data Set Below

To ISPF Library:          Options:
Project . . . _____ Enter "/" to select option
Group . . . . _____ _ Replace like-named members
Type . . . . _____ / Process member aliases
Member . . . _____ (Blank unless member is to be renamed)

To Other Partitioned or Sequential Data Set:
Data Set Name . . . 'UKXXXXX.PARTTWO.JCL (DB2EXTR) '
Volume Serial . . . _____ (If not cataloged)

Data Set Password . . . _____ (If password protected)

To Data Set Options:
Sequential Disposition      Pack Option      SCLM Setting
 1 1. Mod                    3 1. Yes      3 1. SCLM
F1=Help      F2=Split    F3=Exit      F7=Backward  F8=Forward    F9=Swap
F10=Actions   F12=Cancel

MA a 15/055

```

```

Menu Functions Confirm Utilities Help
-----
BROWSE      UKxxxxxx.EXAMPLE.PDS      Row 00001 of 00001
Command ==> _____ Scroll ==> PAGE
Name      Prompt      Size      Created      Changed      ID
r_____  RUNDB2AP
          **End**

F1=Help      F2=Split    F3=Exit      F5=Rfind     F7=Up        F8=Down      F9=Swap
F10=Left     F11=Right   F12=Cancel

MA a 07/003

```

La primera línea de trabajo contiene una instrucción JOB que le asigna un nombre al trabajo.

## 1. Cambia el nombre del trabajo a DB2EXTR.

La siguiente instrucción de trabajo es una instrucción EXEC que marca el inicio de un paso en el trabajo y hace que el programa se ejecute. En este caso, el programa que se ejecutará es IKJEFT01, el Programa Monitor de Terminal de TSO. Este programa ejecutará los comandos especificados en la instrucción de definición de datos SYSTSIN.

El comando DSN conecta la aplicación con un subsistema DB2 en particular.

## 1. Cambia el nombre del subsistema a DB1S.

El comando RUN ejecuta el programa de la aplicación.

1. **Cambia el nombre del programa y el nombre del plan DB2 a DB2EXTR y la librería en donde el programa puede ser hallado a ZOS.CONTEST1.PART2.LOAD.**
2. **Envía el trabajo DB2EXTR y verifica el resultado.**

No parece haber ningún resultado de la ejecución del programa y se ejecutó cero veces. No creo que el programa haya sido ejecutado por el trabajo...



Parece ser que este trabajo contiene un parámetro que solicita al sistema que valide la sintaxis de JCL que escribiste sin que luego pase a ejecutar el trabajo. Este parámetro puede ser útil para verificar errores al escribir o modificar trabajos pero tendrás que eliminarlo y enviar el trabajo una vez más para poder ejecutar el programa.

## 1. Verifica el resultado del trabajo en SDSF para ver si se completó con éxito.

Hm... Más problemas. Tu trabajo todavía está incompleto.



El programa DB2EXTR saca los datos que extrae de la base de datos hacia un archivo llamado ORDERDAT. Sin embargo, éste no es el nombre del data set porque z/OS usa un nombre simbólico de archivo para proporcionar re direccionamiento de nomenclaturas entre el nombre del archivo usado

dentro del programa y el data set real usado durante la ejecución del programa. Este direccionamiento indirecto permite que el mismo programa sea ejecutado sobre diferentes datos sin necesidad de modificarlo o de agregar un soporte de aprobación para diversos parámetros.

La instrucción de definición de datos (DD) de JCL te permite definir un archivo en un paso de trabajo que estará disponible para el programa en ejecución y especificar el data set al que este archivo hace referencia.

Tu trabajo no pudo abrir el archivo de salida ya que no había una instrucción DD correspondiente en el paso de trabajo. Por lo tanto, debes agregar una instrucción DD con un nombre para asignar un data set que el programa pueda utilizar para la salida.

**1. Llama la nueva instrucción DD `ORDERDAT` y especifica los parámetros necesarios de modo que el data set se asigne con los siguientes atributos:**

Usa el parámetro `DSN` para especificar un nombre de data set de `SSAxxxx.HIST.ORDS`  
Usa el parámetro `SPACE` para especificar una asignación primaria y secundaria de espacio de 1 cilindro.  
Usa el parámetro `DISP` para asignar un nuevo data set cuando se ejecute el paso de trabajo y cataloga el data set cuando el paso se haya completado con éxito

Ahora sería un buen momento para consultar el Centro de Información de z/OS, si aún no lo has hecho, bajo la referencia *JCL* en la sección *MVS*.

Te doy una pista: el parámetro `DISP` necesitará dos subparámetros.



**1. Mientras haces esto, responde las preguntas 10 y 11.**

## Una breve explicación sobre administración de almacenamiento

Cuando observes los detalles del parámetro DISP de la instrucción de definición de datos en el libro de Referencia MVS JCL, verás referencias a data sets administrados por SMS. Los data sets en z/OS están distribuidos en volúmenes, del mismo modo que los archivos en tu PC están distribuidos en tu disco duro. Sin embargo, un sistema z/OS tendrá decenas o cientos de volúmenes disponibles.

Esto hace que a los usuarios les sea difícil saber qué volumen usar al asignar nuevos data sets y que a los administradores de almacenamiento les sea difícil equilibrar el espacio libre entre los volúmenes. SMS (Almacenamiento Administrado por el Sistema *System Managed Storage*) resuelve este problema haciendo posible que el sistema seleccione el volumen al cual se deberá asignar un nuevo data set por medio de diversos criterios establecidos por el administrador de almacenamiento. Los data sets asignados de esta forma se llaman data sets administrados por SMS. La mayor parte de los data sets en los sistemas modernos, incluido cualquier data set que asignes, estará administrado por SMS.

También verás referencias a data sets catalogados. El catálogo es el componente de z/OS que te permite localizar el volumen en el que reside un data set con un nombre particular. Esto te permite referirte a ese data set en un trabajo solo por medio de su nombre. Es posible asignar un data set sin agregarlo al catálogo, pero, para poder hacer referencia a un data set no catalogado, necesitarías especificar tanto el nombre del data set como el volumen en dónde reside.

### 1. **Ahora que has hecho esos cambios, envía el trabajo DB2EXTR nuevamente.**

Debe retornar con un código de ceros, lo que indica que ha sido completado exitosamente. Examina el data set de salida para confirmar que el programa ha extraído con éxito los datos de ventas de la base de datos y para familiarizarte con el formato de los datos.

**Nota:** Si ejecutas el trabajo más de una vez, necesitarás borrar primero el data set que se genere. Si no lo haces, surgirá un error de JCL y habrá un mensaje en el resultado del trabajo que indicará que el nuevo data set no pudo ser asignado por causa de un nombre duplicado de data set.

# Introducción a REXX

REXX fue conocido por un tiempo pero sigue siendo muy usado en mainframes. Me gusta porque es muy relajado en cuanto a las reglas de sintaxis: las variables no distinguen mayúsculas de minúsculas, no tiene tipos de datos fijos ni tampoco se deben definir antes de usar. Además, es muy práctico para analizar datos de texto; existen muchas funciones integradas para ello.



REXX (acrónimo de Restructured eXtended eXecutor) es un lenguaje de programación interpretado diseñado por IBM para que tenga una fácil lectura y aprendizaje y al mismo tiempo ofrezca capacidades poderosas. Como es un lenguaje interpretado no hay necesidad de compilar el programa antes de su ejecución; el procesador de comandos REXX procesa directamente cada instrucción de lenguaje tal como la encuentra durante la ejecución del programa. Sin embargo, de haber un compilador REXX disponible sería deseable mejorar el desempeño del programa o la seguridad del código fuente.

Para obtener más información sobre la estructura y la sintaxis del lenguaje REXX y descripciones de los operadores y funciones integradas provistas, consulta las secciones Guía del Usuario de TSO/E REXX y Referencias TSO/E REXX del Centro de Información de z/OS.

Algunos consejos útiles sobre REXX:

- Emite el comando **HILITE REXX** en el editor ISPF para resaltar en color la sintaxis al editar el código REXX.
- Una coma al final de una línea indica que la línea actual de código REXX continúa en la línea siguiente.
- Las variables con el formato  $\alpha.y$  en REXX se llaman variables compuestas y son similares a las matrices. Revisa en la Guía del Usuario de REXX y en las Referencias de REXX los detalles sobre cómo usar estas variables.

## Generación del Reporte a partir de los Datos Extraídos

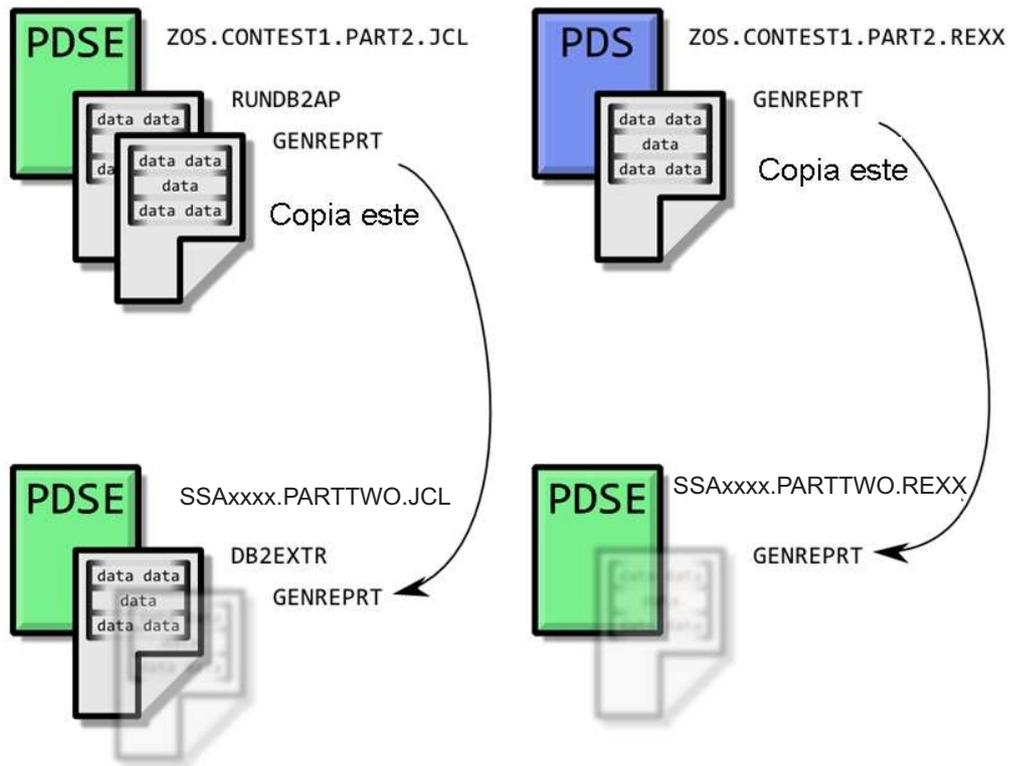
Ahora que extrajiste los datos de ventas de la base de datos, debes ejecutar otro programa para generar el informe. ¡Esta vez puedo darte el programa y el trabajo JCL!



REXX es un lenguaje muy apropiado para la tarea de generación de informes como estos debido a su

capacidad de manipulación de caracteres.

1. **Copia el trabajo desde ZOS.CONTEST1.PART2.JCL(GENREPT) a tu librería JCL y el código fuente del programa desde ZOS.CONTEST1.PART2.REXX(GENREPT) a tu librería REXX.**



Notarás que este trabajo tiene dos pasos. El primer paso borra el data set del informe en caso de haber uno, lo cual es muy útil si necesita enviar el trabajo más de una vez.

1. **Reemplaza los marcadores de posición ++XXXX++ adecuadamente en el trabajo, nombra tu data set de informes como SSAxxxx.SALES.REPORT y envíalo.**

Yo dije que te daría el trabajo JCL, ¡pero nunca dije que funcionaría la primera vez!



**1. Corrige el error y vuelve a enviar el trabajo.**

Parece que el programa REXX tampoco funcionó la primera vez. No percibimos esto antes porque REXX es interpretado: no había llegado a este punto cuando hallamos el primer error. ¡Si lo hubiéramos compliado, lo habríamos descubierto!



**1. Corrige el error en el programa y otros errores que puedas encontrar.**



Tu informe dice que los artículos que más se venden tienen menos ventas que los que menos se venden. ¡Este informe está ordenado al revés! ¡Corrígelos!

Esta vez deberá ejecutarse sin errores y se deberá generar un informe de ventas. Sin embargo, no parece que el informe de los artículos de mayor y menor venta haya sido ordenado correctamente.

**1. Corrige este problema en el código REXX y vuelve a enviar el trabajo.**

## Mejora tu JCL

¡Buen trabajo! ¡Arreglaste el JCL y el REXX y lograste generar el informe que el jefe quería!  
Pero calculo que puedes mejorar el JCL mucho más: ¿puede combinar este trabajo con el trabajo DB2EXTR que usamos antes?



En lugar de tener que enviar dos trabajos por separado para extraer los datos de venta de la DB2 y generar el informe, todo esto puede hacerse en un trabajo con dos pasos (tres pasos si incluyes el primer paso que borra el data set actual de informes).

### 1. **Modifica tu trabajo GENREPRT de tal modo que realice ambas tareas.**

Podría ser útil modificar además el primer paso del trabajo para que borres el informe antiguo y los data sets de ventas extraídas.

El data set con las ventas extraídas de la base de datos se transfiere al segundo paso, el cual genera el informe. No es necesario guardarlo una vez terminado el trabajo.



Para este objetivo se puede usar un tipo especial de data set llamado data set temporal. Los data sets temporales se crean y se borran con un único trabajo.

1. **Modifica el data set usado para almacenar los datos de venta extraídos de la base de datos para convertirlo en un data set temporal.**
2. **Mientras haces esto, responde las preguntas 12 y 13.**

**Consejo:** Deberás cambiar el nombre del data set para que respete un formato determinado ya que esto es lo que le indica al sistema que el data set es temporal. Además, deberás cambiar la disposición del data set de modo tal que el data set asignado en el primer paso sea transmitido para ser usado en el segundo paso.

## Envía tus Respuestas

1. **Revisa tus respuestas y luego envíalas** para que puedan ser calificadas. Tu informe también será revisado pero no tienes que enviarlo de la misma manera. **Asegúrate, sin embargo, de que contenga datos de todo el año y no de un mes solamente.**

Para hacer esto, vé a la opción 6 (Command - Enter TSO or Workstation) e ingresa:

```
XMIT ZOS19.MATTK DA('SSAxxxx.SSAxxxx.ANSWERS')
```



¡Felicitaciones! Has completado todos los pasos de la Parte Dos! Ahora podemos direccionar nuestra publicidad hacia los productos más populares que vendemos.

¡Has logrado mucho hasta aquí, pero queda más por hacer! La Parte Tres agrega lo que has venido aprendiendo y exige aún más esfuerzo. Le agregarás funcionalidad al sitio Web usando otro de los productos IBM famosos en todo el mundo.



Voy a recomendarte a nuestro equipo de eventos para la Parte Tres de la prueba; vas a necesitar acceso a sus sistemas de mainframe. Envía un correo electrónico a [inicaad@ar.ibm.com](mailto:inicaad@ar.ibm.com) para solicitar una nueva identificación de usuario, en el campo de asunto escribe "Solicitud *tu\_id\_actual* ID etapa 3" .  
¡Buena suerte!