

---

---

# SISINF

**BATCH**

**Versión 7.3**

---

---

**Kratos**, S.A. de C.V.

La Tecnología en Software.

Derechos Reservados ©. Prohibida la reproducción total o parcial sin permiso escrito de **KRATOS, S.A. de C.V.** El uso de programas que integran **SISINF** se vende y renta bajo contrato con **KRATOS, S.A. de C.V.**

## CONTENIDO

---

CONTENIDO.....	2
PREFACIO.....	3
1) Introducción.....	4
2) Ejecución de los Módulos de SISINF en BATCH.....	6
3) Programa SIS/B.....	8
4) Opciones *M, *C y *D de SIS/B.....	10
5) Ejemplos de uso de SIS/B.....	13
6) Datos para ELS con *D.....	16
7) Errores al llamar a SIS/B.....	18
8) Cancelación de la ejecución en BATCH.....	19

## PREFACIO

---

En el desarrollo de aplicaciones administrativas, siempre hay procesos largos cuya ejecución es conveniente en horarios especiales o bajo condiciones diferentes a los trabajos interactivos. La opción de BATCH de SISINF permite ejecutar los módulos en esta forma, además de contar con las opciones necesarias para hacer estos programas independientes de sistema operativo ya que normalmente esta forma de trabajo implica estatutos de control específicos en cada ambiente.

## 1) Introducción.

---

En la operación normal de una computadora, cuando ya se tienen varias aplicaciones en producción, es común que en un momento dado se estén ejecutando programas interactivos y programas de proceso. Entendiéndose por programas interactivos aquellos hechos para captura o consulta, y por programas de proceso aquellos cuya función es sacar listados, afectar archivos, hacer algún tipo de cálculo, etc.

Al tener esta mezcla de programas, se pueden tener las siguientes situaciones:

- ◆ Si ambos tipos de programas tienen la misma prioridad de ejecución dentro del Sistema Operativo, el resultado es que los tiempos de respuesta de los programas interactivos **sea inadecuado**.
- ◆ Usar una terminal para la ejecución de programas de proceso, en que ordinariamente NO se lee información o se lee al inicio, y necesitar la terminal para ejecución de programas interactivos.
- ◆ Debido a la memoria que se tenga disponible, puede suceder que sea **suficiente** para la ejecución de programas interactivos, pero **insuficiente** si se tiene una mezcla de varios programas interactivos con varios de proceso. Lo anterior se puede deber a que dicha mezcla produzca niveles de paginación o "swapping" inadecuados.

En el **desarrollo** de sistemas es usual que la actividad del programador sea usar el editor para corregir sus programas, usar **CLS** para compilarlos y usar **ELS** para probarlos. Pero en sistemas de producción en que por necesidades de la aplicación se cambia una tabla o un archivo, es usual que se necesiten compilar muchos programas, uno después de otro, siendo éste un caso igual al de **programas de proceso**.

El problema descrito anteriormente es un problema de recursos, entendiéndose por éstos la unidad central de proceso (cpu), su memoria, acceso a discos, etc. Y el problema consiste en que si en un momento dado todos los usuarios de la computadora necesitan de uno de dichos recursos, al compartirse entre muchos se tendrá una degradación del sistema. Dicho de otra forma, si todos los usuarios de la computadora tanto de desarrollo de sistemas como los usuarios de éstos, en cualquier tiempo puede ejecutar cualquier programa, se podrá tener en ciertos momentos una degradación del sistema.

La solución al problema anterior son varias, entre otras se puede mencionar:

- ◆ Establecer políticas de operación para ejecutar ciertos procesos en ciertas horas. Así se puede diferir para un turno nocturno ciertos procesos de actualización y listados.
- ◆ Adecuar los recursos de memoria, acceso a discos, etc. a las necesidades de la instalación.

En muchos sistemas operativos se tiene la posibilidad de ejecutar programas en BATCH teniendo las siguientes ventajas:

- ◆ El sistema operativo asigna una prioridad menor a los programas en BATCH, dando por resultado que los demás programas tengan tiempos de respuestas mejores.
- ◆ Regular el número de programas BATCH que se ejecutan en forma simultánea evitando que en un momento dado se tengan muchos programas de proceso consumiendo recursos de CPU, memoria, etc.
- ◆ No tener que usar una terminal para la ejecución de procesos BATCH.

Los módulos de **SISINF** (DBD, CLS, ...) se pueden ejecutar en forma interactiva o bien en BATCH. Se tiene además el programa SIS/B que permite en forma dinámica por medio de ELS ejecutar programas en BATCH.

En este Manual se describe el uso de los módulos de **SISINF** en BATCH así como el programa SIS/B. Esta **facilidad es opcional en la instalación.**

Se recomienda ver el Manual de Implantación sobre características y restricciones del BATCH y de SIS/B en la máquina en que se trabaja.

## 2) Ejecución de los Módulos de SISINF en BATCH.

---

Cuando se dispone de la operación BATCH en una instalación, todos los módulos de **SISINF** están preparados para poder usar esta facilidad del sistema operativo.

**En forma general**, los pasos a seguir para usar esta facilidad del sistema operativo son:

- ◆ Formar con el editor un archivo que contenga las mismas instrucciones y datos que se teclearían para ejecutar un módulo de **SISINF** en una terminal, teniendo en cuenta que cada lectura de la terminal es un registro del archivo editado.
- ◆ Ejecutar la instrucción del sistema operativo en la que se indica que en dicho archivo se tiene un trabajo BATCH. Al hacer lo anterior el sistema operativo agrega este nuevo trabajo a su lista de trabajos pendientes de procesar en BATCH.
- ◆ Cuando se ejecuta el trabajo, los datos **que se leían** de la terminal se leen del archivo y lo que se escriba en la terminal se lista.

Se recomienda que se estudien los Manuales del Sistema Operativo en donde se describa la opción BATCH.

Dado que la misma información que se tecléa en la terminal, debe estar en el archivo, algunas consideraciones sobre la ejecución en BATCH de los módulos de **SISINF** serían:

- ◆ Para ejecutar los compiladores (DBD, ...) se recomienda pedir el listado por Impresora ya que si se usa la opción C (CRT) será necesario calcular los registros en blanco para la lectura del SIG. al llenarse la 'pantalla'.
- ◆ Al ejecutarse los compiladores si se pide el listado por Impresora y luego se pide la de **Errores** y **NO** hay errores, NO se generará el listado.
- ◆ En el caso del módulo ELS, es necesario dar un registro en blanco para la lectura del **primer** SIG., en los demás no se efectúa dicha lectura al llenarse la 'pantalla'.
- ◆ Si hay errores al ejecutarse el módulo ELS, se asume EJ para diagnósticos informativos, y FI para errores de cancelación.

Un ejemplo del archivo editado para compilar un programa en un supuesto sistema operativo, podría ser:

# de Registro	Contenido
(1)	:JOB
(2)	:RUN CLS
(3)	xxxxx
(4)	Z1
(5)	I
(6)	E
(7)	:END

El primer registro tiene la instrucción del sistema operativo para iniciar trabajos en BATCH, en el segundo se pide la ejecución de CLS, en el tercero se da la clave de acceso, en el cuarto se da las iniciales del programa asumiendo que la clave de acceso tiene prefijados los demás valores, por último se pide Impresora, Errores y se termina el trabajo.

Un ejemplo de ejecución del programa previamente compilado sería:

# de Registro	Contenido
(1)	:JOB
(2)	:RUN ELS
(3)	xxxxx
(4)	Z1
(5)	
(6)	090996
(7)	:END

El registro 5 es para la respuesta a la lectura del primer SIG., se asumió que el programa sólo lee una fecha, la cual se especifica en el registro 6.

### 3) Programa SIS/B.

---

El programa SIS/B permite enviar trabajos al BATCH al estar ejecutando el módulo ELS.

Para ello es necesario escribir en un archivo temporal las mismas instrucciones y datos que se teclearían para ejecutar un módulo de **SISINF** en la terminal. Una vez hecho lo anterior por medio de la instrucción de LLAMAR se pide la ejecución de SIS/B. El programa SIS/B toma el archivo temporal y le pide al Sistema Operativo que ponga en su lista de trabajos BATCH el del temporal.

Las instrucciones de **SISINF** para el último ejemplo de la sección anterior serían:

```
OPCION TEMPORAL 950 ESCRITURA ASCII
ESCRIBIR TEMPORAL 950 ':JOB'
ESCRIBIR TEMPORAL 950 ':RUN ELS'
ESCRIBIR TEMPORAL 950 'XXXXXX'
ESCRIBIR TEMPORAL 950 'Z1'
ESCRIBIR TEMPORAL 950 '_'
ESCRIBIR TEMPORAL 950 ':END'
LLAMAR 'SIS/B_' 950
```

Un ejemplo de la utilización de SIS/B podría ser el caso de cuando se desea sacar un catálogo de un archivo grande y no se quiere afectar los tiempos de respuesta de los programas interactivos. Para lograrlo, el programador de la aplicación pondría instrucciones parecidas a las del párrafo anterior en el programa de MENU, es decir, las instrucciones para ESCRIBIR el temporal y la de LLAMAR, y el usuario al pedir dicho catálogo mandaría su programa al BATCH, sin necesidad de saber las instrucciones para hacerlo.

Otro ejemplo de la utilización de SIS/B podría ser generar los respaldos para el caso de la nómina. En esta forma el usuario al final del cálculo ejecuta un programa de ELS que genera en un temporal las instrucciones del Sistema Operativo para hacer el respaldo y llamar a SIS/B, teniéndose por ventaja el hacer este proceso más seguro.

Hasta ahora los ejemplos vistos escriben en el archivo editado o en el temporal, instrucciones para un Sistema Operativo **supuesto**, por lo que será necesario que el programador estudie **qué** instrucciones son necesarias para

poder utilizar el BATCH en la máquina en que él trabaja.

Es necesario aclarar que al poner en un temporal instrucciones específicas de un sistema operativo, hace que el programa de **SISINF NO SEA TRANSPORTABLE A OTRAS COMPUTADORAS.**

Dado los **problemas descritos**, es decir, el tener que estudiar las instrucciones para el BATCH y la NO transportabilidad de los programas de **SISINF**, el programa SIS/B ofrece lo siguiente:

- ◆ Ciertas instrucciones propias de SIS/B que generan instrucciones específicas al Sistema Operativo que se use.
- ◆ La posibilidad de pedir la ejecución de SIS/B con el nombre XXX/B en donde XXX son las iniciales del Sistema Operativo que se usa.

Es decir, si se pone XXX/B o SIS/B se llamará al **mismo** programa cuyo nombre real es diferente, pero al poner XXX/B se asume que las instrucciones especificadas en el archivo temporal son para el Sistema Operativo con iniciales 'XXX' y al poner SIS/B se asume que serán instrucciones compatibles entre Sistemas Operativos.

En esta forma, si el archivo temporal generado en un programa de **SISINF** tiene instrucciones específicas de un sistema operativo cuyo nombre es XYZ y se pone en la instrucción de llamar XYZ/B, al tratar de ejecutar dicho programa en otra computadora **diferente** se tendrá el diagnóstico de **programa inexistente**, avisando en esta forma al programador que es necesario cambiar dichas instrucciones.

En resumen, el hecho de llamar a XXX/B **indicará** que se están usando instrucciones de un sistema operativo específico.

Por último, es necesario conocer ciertas consideraciones en la forma de crear el temporal-

- ◆ El temporal debe ser ASCII, es decir, en OPCION TEMPORAL se puso esta opción.
- ◆ La longitud mayor de registro debe ser menor o igual a 80 caracteres o lo que pida el Sistema Operativo.
- ◆ El temporal formado se borra al llamar a SIS/B o XXX/B.

## 4) Opciones \*M, \*C y \*D de SIS/B.

---

Las opciones descritas en esta sección sirven para que el programador pueda hacer lo siguiente:

- ◆ Poner instrucciones en el temporal que sean compatibles entre Sistemas Operativos.
- ◆ Complementar instrucciones de un Sistema Operativo específico con 'password', clave de acceso, ...

Cuando se desea pedir la ejecución de algún módulo de **SISINF** se pueden poner las instrucciones propias del Sistema Operativo en que se trabaje o bien usar la opción de \*M. Dicha opción consiste en poner en un registro del temporal la clave \*M, el nombre de módulo y los datos necesarios para su ejecución. El programa SIS/B genera con esta información las instrucciones y datos que se necesitan en el Sistema Operativo que se trabaja. En esta manera, la forma de pedir la ejecución de un módulo de **SISINF** es compatible entre Sistemas Operativos.

El contenido que debe tener el registro escrito en el archivo temporal es en formato general:

```
*M   aaa   ,   b   ,   c   ...
```

En donde:

- ◆ \*M en las dos primeras columnas le indica a SIS/B que se desea pedir la ejecución de un módulo de **SISINF** y que transforme la información que se especifica a continuación en las instrucciones necesarias en el Sistema Operativo que se está trabajando.
- ◆ aaa es el nombre del módulo de **SISINF** separado por uno o más blancos de la clave \*M.
- ◆ b, c, ... son los datos que necesita el módulo para su ejecución, separados por comas, pudiendo tener blancos intermedios y omitiendo la clave de acceso. Así por ejemplo, los datos para DBD son iniciales de DBD, unidad de disco, se lista por CRT o Impresora y se lista la DBD o errores.

Los siguientes comentarios y restricciones son aplicables a esta opción de SIS/B.

- ◆ Esta opción es válida para los módulos DBD, CLS, ELS y RECA.
- ◆ Para el módulo ELS, si el programa cuya ejecución se pide lee información, esta se deberá especificar por medio de la opción \*D.
- ◆ Para el módulo RECA, al menos las iniciales de DBD, el dispositivo de DBD y el nombre del archivo, deben ser especificados en la instrucción \*M, los dispositivos de los archivos de entrada y salida, se asumen blancos si no se especifican; Si se desea ejecutar alguna opción de RECA además de reconstruirlo se deberá especificar también en la instrucción de \*M, y sus datos deberán pasarse a través de instrucciones de \*D.
- ◆ No se deberá de especificar la clave de acceso ya que SIS/B pone la misma que se dio al ejecutar ELS.
- ◆ Se deben de especificar todos los datos que necesita el módulo independientemente de los valores prefijados que tenga la clave de acceso. El programa SIS/B revisa qué valores tiene prefijados y NO pone esta información. Dado que el módulo ELS, al ejecutarse en BATCH siempre lee las INICIALES DE PROGRAMA aunque se tenga un valor prefijado, SIS/B siempre deja este dato.
- ◆ En los compiladores si no se especifica la unidad del listado se asume Impresora, y si no se especifica que listar, se asume Errores.

La opción de \*D sirve para indicarle a SIS/B que la información del registro es un dato para un módulo de **SISINF**, es decir el programa SIS/B transforma la clave \*D a lo que sea necesario para indicarle al Sistema Operativo en que se está trabajando que la información es un dato para el módulo.

En casi todos los Sistemas Operativos, es necesaria poner cierta instrucción al inicio que indique los datos del usuario que pide el trabajo BATCH, y otra al final para indicar que ya no se tiene más información.

La opción \*C sirve, entre otras cosas, para generar estas instrucciones al poner:

\*C \*INICIO\*

\*C \*FIN\*

Es decir, al encontrar SIS/B la clave de \*C procede a convertir la palabra que se indique entre asteriscos. En este caso, con \*INICIO\* genera la o las instrucciones del inicio, y con \*FIN\* genera la instrucción al final que requiere el Sistema Operativo en que se está trabajando.

Se recomienda revisar el Manual de Implantación, en la sección de SIS/B para opciones específicas de \*C en la máquina que se está trabajando.

## 5) Ejemplos de uso de SIS/B.

El siguiente ejemplo es un programa de **SISINF** que leyendo las INICIALES DE PROGRAMA de varios programas, genere un trabajo BATCH para compilarlos:

a) En DBD se puso:

AREA

```
X2      * * R0  S  X  2
X40     * * R0  S  X 40
```

b) El programa sería:

```
PROGRAMA 'COMPILAR VARIOS PROG DE DBDSL'
MOVER '*M CLS, LL,, XX' A X40
OPCION TEMPORAL 950 ESCRITURA ASCII
ESCRIBIR TEMPORAL 950 '*C *INICIO*'
1 LEER TERMINAL EJECUTA 3 $
'INICIALES DE PROGRAMA' X2
CONVERTIR X2 A X40 /POS. 1 12 /LONG. 2
ESCRIBIR TEMPORAL 950 X40
EJECUTA 1
ESCRIBIR TEMPORAL 950 '*C *FIN*'
LLAMAR 'SIS/B' 950
FIN
```

En este programa se generan en el temporal 950 con '\*C \*INICIO\*' la o las instrucciones de inicio del trabajo en BATCH, luego con la opción \*M se generan las instrucciones para compilar los programas, y al final se termina con '\*C \*FIN\*'

El siguiente ejemplo es la ejecución de un programa de **SISINF** por ELS en BATCH. Para hacer más completo el ejemplo, se supone que dicho programa lee una fecha.

a) En DBD se puso:

AREA

```
FECHA      *  * R0  S  F
X12        *  * R0  S  X  12
PREG*      * R0  S  S
```

b) El programa sería:

```
_____
_____
_____
_____
```

Desplegar las opciones del menú y leer una.

\*

\* EJECUCION DE PSLLA5

\*

100 LEER TERMINAL EJECUTA 1 \$

```
'EN BATCH SI O NO' PREG
SI PREG = 'SI' EJECUTA 101
FIN 'LLA5'
```

101 LEER TERMINAL EJECUTA 1 \$

```
'FECHA DEL PROCESO' FECHA
```

```
_____
```

\_\_\_\_\_ Instrucciones para validar la fecha contra archivos

```
_____
```

```
MOVER '*D' A X12
```

```
CONVERTIR FECHA A X12 /POS. 7 4 /LONG. 2
```

```
CONVERTIR FECHA A X12 /POS. 5 7 /LONG. 2
```

```
CONVERTIR FECHA A X12 /POS. 3 10 /LONG. 2
```

```
OPCION TEMPORAL 970 ESCRITURA ASCII
```

```
ESCRIBIR TEMPORAL 970 '*C *INICIO*'
```

```
ESCRIBIR TEMPORAL 970 '*M ELS ,LL,A5'
```

```
ESCRIBIR TEMPORAL 970 X12
```

```
ESCRIBIR TEMPORAL 970 '*C *FIN*'
```

```
LLAMAR 'SIS/B' 970
```

```
EJECUTA 1
```

```
_____
```

---

En el programa de MENU se supone que las primeras instrucciones desplegarán las opciones del mismo y el usuario selecciona qué ejecutar.

Primeramente se pregunta si será en BATCH, en caso negativo se llega a FIN 'LLA5' y dicho programa se ejecutará por la terminal. En caso afirmativo, será necesario leer la fecha del proceso y ponerla como dato en el archivo temporal, ya que el programa PSLLA5 lee dicha fecha, los pasos seguidos son:

- a) Validar que dicha fecha sea una fecha correcta posiblemente contra información de archivos.
- b) Convertir en X12 la fecha, de forma que si el usuarios teclea 9 9 05, en X12 se tendrá '\*D\_09\_09\_05'.
- c) Se genera el temporal en que se pide la ejecución de ELS con el dato de la fecha.
- d) Se llama a SIS/B.

## 6) Datos para ELS con \*D.

Cuando se pasan datos al programa de **SISINF** que se ejecutará en BATCH es necesario tomar en cuenta las consideraciones dadas a continuación en el programa que pide la ejecución. Las consideraciones son:

- ◆ Las variables N1, N2, N3, N4 y F deberán de leerse como tales y por medio de la instrucción CONVERTIR pasar al formato de lectura. (Ver ejemplo en sección anterior).
- ◆ Si las variables N1, N2, N3 y N4 son con cero decimales y positivas se pueden escribir directo en el temporal.
- ◆ Las variables X y S se deben pasar directo en el temporal.

El siguiente ejemplo ilustra estos casos:

a) En DBD se puso:

```
X8          * * R0  S  X  8
NOMB        * * R0  S  X  30
DATO1       * * R0  S  N1  2
DATO2       * * R0  S  N1  0 10 10 R 1 1000
```

b) Un programa que lea DATO1, DATO2 y NOMB y lo pase al programa a ejecutarse en BATCH sería:

```
LEER TERMINAL EJECUTA 10 DATO1 DATO2 NOMB
MOVER ' ' A X8
SI DATO1 > 0 EJECUTA 1
CONVERTIR '-' A X8 POS. 1 1 EJECUTA 5
1 CONVERTIR DATO1 A X8 /POS. 1 2 /LONG. 2
CONVERTIR '.' A X8 POS. 1 4 EJECUTA 5
CONVERTIR DATO1 A X8 /POS. 3 5 /LONG. 2
ESCRIBIR TEMPORAL 950 '*D' X8
ESCRIBIR TEMPORAL 950 '*D' DATO2
ESCRIBIR TEMPORAL 950 '*D' NOMB
```

c) A continuación se ilustra la información que se genera en el temporal 950

con ciertos valores de las variables.

DATO1 = 5.6

DATO2 = 85

NOMB = PRUEBA

\*D\_05.60\_\_

\*D0085

\*DPRUEBA

## 7) Errores al llamar a SIS/B.

---

Si al procesar el archivo temporal, el SIS/B detecta algún problema o tiene error al enviar a ejecutar el trabajo, regresa a ELS un código indicando la causa, éstos pueden ser:

- 1XXX Error XXX en temporal.
- 2XXX Error XXX en SBIXX o SBDXX.
- 3001 El temporal no es ASCII.
- 3002 Longitud del temporal no está entre 1 y 80.
- 3003 Información entre asteriscos no está en tabla.
- 3004 Error Interno de ELS-SISB, Código de instrucción ilegal.
- 3005 Módulo que se pide ejecutar no está en tablas.
- 3006 Registro de salida de más de 100 columnas.
- 3007 Se pide DBD-C o CLS-C y no se puede cancelar o tiene prefijados.
- 3008 Ya existen todos los archivos BRXXYY (yy=00-99)
- 4XXX Error XXX al pasar SBIXX al sistema operativo.

## 8) Cancelación de la ejecución en BATCH.

---

Todos los módulos de **SISINF** cuando se ejecutan en BATCH, pasan al Sistema Operativo el estado de su terminación al finalizar su ejecución el cual puede ser:

- ◆ Normal.
- ◆ Errores informativos.
- ◆ Errores de cancelación.

Dependerá del Sistema Operativo la forma en que este estado se puede consultar (en el Manual de Implantación se detalla como hacerlo).

Lo anterior sirve para que en cierto tipo de procesos, en que se ejecuta varias veces el módulo ELS y luego posiblemente el módulo RECA, se pueda **cancelar** la ejecución si hay error en cualquiera de los pasos.

El módulo ELS tiene los siguientes cambios cuando se ejecuta en BATCH.

- ◆ Si ocurren errores informativos, asume se contesta con la opción de EJ (Continuar Ejecución) y al final pasará dicho estado.
- ◆ Si ocurren errores de cancelación, asume se contesta con la opción de FI (Fin) y termina la ejecución reportando dicho estado.
- ◆ Si se ejecuta la instrucción de ERROR, se reportará al terminar la ejecución el **estado** de cancelación.

Con esta opción se pretende que si el programador detecta una situación de cancelación, mande el ERROR y termine (FIN).

Note que para hacer uso de estas opciones es necesario conocer las instrucciones del Sistema Operativo para saber el estado de terminación y poner las instrucciones necesarias en el procedimiento para hacer la cancelación.