UNIDAD

13

DIPLOMATURA EN PROGRAMACION ABAP

MÓDULO 13: BATCH INPUT

BATCH INPUT

Este módulo introduce los conceptos relacionados con la importación de datos desde un sistema *legacy* as **SAP** mediante la técnica denominada *Batch Input*.

Introducción

Cuando se instala una aplicación en productivo es necesario dar de alta toda la información indispensable para que la empresa pueda funcionar (proceso de migración de dalos o conversión).

Por ejemplo, antes de poder generar facturas reales será necesario introducir todos los clientes activos y todos los productos que están a la venta.

Para realizar la carga de productos que están a la venta se debería ejecutar manualmente la transacción "Alta de material" tantas veces como productos tengamos y la misma operación con "Alta de clientes" para todos los clientes. En el caso de que la empresa tenga muchos productos y muchos clientes, la carga inicial será muy costosa.

Generalmente todos estos datos maestros (clientes, materiales, proveedores) ya están en el antiguo sistema informático. Por lo tanto lo ideal será disponer un mecanismo que nos permitiese trasladar los datos de un sistema a otro.

A la hora de la migración de datos de un sistema externo a SAP, tenemos dos posibilidades:

- Realizar programas que llenen todas las bases de datos SAP involucradas, mediante instrucciones directas de SAP-SQL.
- Utilizar la técnica del Batch Input de SAP.

Para muchas transacciones, la primera de las opciones es inviable, debido a la complejidad de la estructura de datos SAP y para mantener la integridad de la misma la cantidad de validaciones que se deberían realizar sobre los datos de entrada sería enorme. Como consecuencia, tanto el coste en diseño, codificación y pruebas sería altísimo.

En cambio, la técnica de los Batch Input de SAP nos permite realizar todas las verificaciones automáticamente, con un coste en diseño y desarrollo mínimo. En este capítulo veremos cómo utilizar la técnica de los Batch Input.

Un Batch Input es un método seguro y fiable de transferir datos hacia un sistema SAP. Suele utilizarse cuando deben realizarse un elevado número de altas, modificaciones o borrados.

Para garantizar la integridad del sistema, los datos son sometidos a los mismos controles de validación y a las mismas operaciones de base de datos en SAP como si fueran introducidos manualmente y uno por uno, por el usuario. Es decir realmente la técnica del Batch Input consiste en simular repetidamente un proceso online (transacción), durante un proceso Batch.

El proceso de carga de datos se realiza en dos fases:

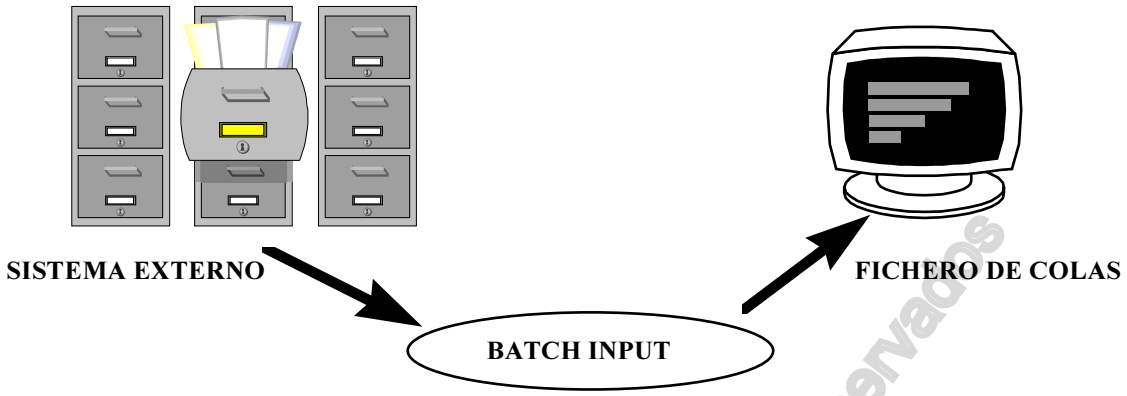
- Fase de Generación: A partir de una fuente de información como puede ser un fichero de entrada, donde estarán todos los datos que queremos cargar en SAP, se transformaran estos datos en un formato determinado, para almacenarlo en una estructura de SAP que llamaremos fichero de colas.

- Fase de Proceso: A partir de la información grabada durante la fase de generación en el fichero de colas, se cargarán los datos físicamente en la base de datos.

Con la técnica del Batch Input, se realiza una simulación del diálogo del usuario con la máquina, es decir haremos exactamente lo mismo con la única diferencia de que la entrada de datos en vez de ser manual, será automática a partir de un fichero de colas.

Fase de generación del Batch Input.

En esta fase se realiza la transferencia de los datos de un sistema externo a un fichero de colas. Para ello se debe codificar un programa de Batch Input.



Sistema externo

La extracción de los datos de un sistema externo suele ser realizada por el departamento de informática de la empresa donde va a ser instalado SAP, ya que es quien mejor conoce la estructura de su actual sistema informático. Normalmente el resultado final de esta extracción de datos será un fichero secuencial con los datos necesarios para cargar en SAP. El programa Batch Input, leerá este fichero y transformará los datos a un formato determinado para poder almacenarlos en el fichero de colas.

El fichero secuencial tendrá una estructura de registro que deberá ser conocida por el equipo de desarrollo de SAP. Generalmente, y siempre que sea posible, se asociará un registro a una transacción de datos SAP. Por ejemplo, en el caso de altas de materiales, en un registro se guardarán todos los datos necesarios para dar de alta un único material.

Por regla general, el sistema externo es un fichero secuencial en el que se encuentran los datos con los que se desean simular las transacciones. No obstante no tiene que ser necesariamente un fichero secuencial, sino que puede ser cualquier fuente de información que tengamos (tablas físicas de SAP, tablas de otras bases de datos relacionales, etc.).

El programa Batch Input

Es el único desarrollo que se debe hacer en ABAP/4.

El programa de Batch Input leerá el fichero secuencial y transformará los datos a un formato determinado, para almacenarlos en una entrada del fichero de

colas. Dichas entradas se denominan sesiones. Cada programa de Batch Input genera una sesión. Estas sesiones pueden contener una o múltiples transacciones.

Una transacción en SAP consta de una serie de pasos de diálogo. El programa de Batch Input debe preparar los datos para cada uno de los pasos de diálogo de la transacción.

Por ejemplo, imaginemos que para dar de alta un material el sistema ejecuta una transacción de tres pantallas:

- Pantalla 1: Entrada de los datos sobre el diseño de material (peso, altura, volumen...).
- Pantalla 2: Entrada de los datos sobre ventas del material (precio, descuentos.
- Pantalla 3: Entrada de los datos sobre la producción (costes, almacenaje).

El programa que genere la sesión de altas de materiales deberá por tanto, programar la secuencia de acciones y pantallas en el mismo orden que la transacción y preparar los datos en cada una de estas pantallas, para cada material que se quiera dar de alta. Por ello, antes de programar un Batch Input es necesario un conocimiento exhaustivo de la transacción que se desea simular, puesto que ganaremos mucho tiempo si estudiamos previamente el funcionamiento de ésta.

Cómo se codifica un Batch Input lo veremos más adelante.

El resultado de esta etapa será una sesión de Batch Input grabada en un fichero y que posteriormente deberá procesarse para cargar físicamente los datos en el sistema SAP.

El fichero de colas

Todos los programas Batch Input graban entradas (sesiones) en el fichero de colas. Para posteriormente poder identificar cual es la sesión que nos interesa procesar, las sesiones poseen un formato determinado:

- Nombre de la sesión.
- Usuario que ha creado la sesión.

- Mandante en el que debe procesarse.
- Número de transacciones que contiene.
- Número de pantallas que contiene.
- Datos adicionales.

Una sesión de Batch Input puede encontrarse en uno de los siguientes estados:

- A procesar: Si la sesión todavía no ha sido procesada.

- Procesada: Si las transacciones que componen la sesión han sido ejecutadas íntegramente sin errores.

- Erróneas: Si en la sesión aún quedan transacciones que no se han procesado correctamente. Cuando una sesión está en estado incorrecto, no quiere decir que las transacciones que contenía no hayan sido procesadas, sino que algunas se han procesado y otras no. Estas transacciones erróneas las podremos reprocesar más adelante, es decir nunca perdemos una transacción a no ser que explícitamente borremos la sesión.

- Siendo creada: Si hay un programa Batch Input que está generando una sesión en ese momento.

- En proceso: Si se está procesado en ese instante la transacción.

- Fondo: Si se ha lanzado la sesión para que se procese pero todavía no ha comenzado a ejecutar por falta de recursos del sistema.

Fase de procesado de una sesión.

Para gestionar el fichero de colas utilizaremos la transacción SM35 (Sistema → Servicios → Batch Input → Tratar).

Mediante esta transacción podemos consultar, eliminar y procesar todas las sesiones de Batch Input.

Una vez generada la sesión con el programa Batch Input, accederemos a la transacción SM35 y marcaremos la sesión que nos interesa procesar.

Existen 3 tipos de procesamiento:

- Procesar visible.
- Procesar visualizando sólo errores.
- Procesar en invisible.

Durante la ejecución de una sesión se irá grabando en un "log" de proceso, el resultado de cada transacción. Entre la información que nos ofrece el log destaca:

- Hora de inicio de proceso de la sesión.
- Hora de inicio de proceso de cada transacción.
- Mensajes de incidencia o de proceso correcto (los mismos que daría la transacción en el caso de ejecutarla manualmente).
- Estadística final de proceso:
- Nº Transacciones leídas.
- Nº Transacciones procesadas con éxito.
- Nº Transacciones erróneas.

Siempre que existan transacciones con errores se podrán reprocesar.

- Procesamiento Visible: Con este método se procesa cada una de las transacciones visualmente, es decir, el usuario va visualizando todas y cada una de las pantallas que hemos programado. El usuario únicamente debe ir pulsando <intro> para saltar de una pantalla a otra. Asimismo, si se cree conveniente, se permite modificar los valores de algún campo de la pantalla.

Si una transacción no interesa procesarla, podemos cancelarla (pudiendo ser ejecutada con posterioridad) o podemos borrarla (no se podrá ejecutar). Todas las transacciones que cancelemos se grabarán en la sesión y la sesión pasará a estar en estado incorrecto.

No devuelve el control del sistema al usuario hasta que todas las transacciones hayan sido procesadas o cancelemos el Batch Input.

- Procesamiento Invisible: El sistema procesará en fondo batch la transacción. Es decir, toda la ejecución es transparente al usuario. El usuario recupera el control del sistema inmediatamente. Para ver el resultado de la ejecución de una sesión, tendrá que ver el "log" de proceso una vez haya finalizado.

- Procesamiento visualizando sólo errores: El sistema procesará cada una de las transacciones en modo invisible hasta que detecte un error, en cuyo caso parará el proceso en la pantalla donde se ha producido el error, pudiendo entonces el usuario detectar y corregir dicho error o cancelar la transacción. Una vez corregido el error o cancelada la transacción, el sistema continúa procesando el resto de transacciones.

No devuelve el control del sistema al usuario hasta que todas las transacciones hayan sido procesadas o cancelemos el Batch Input.

Consejos prácticos en la utilización de Batch Inputs.

Para conocer el código de la transacción, el nombre de las pantallas de cada transacción y los nombres de los campos que se desean completar haremos lo siguiente:

- Código de la transacción: Entrar en la transacción a simular e ir a Sistema → Status.

- Nombre de la pantalla: Una vez estarnos en la pantalla que necesitamos, hacemos lo mismo que en el punto anterior, anotando el programa (Dynpro) y el número de dynpro.

- Nombre de los campos: Una vez situados sobre el campo en cuestión, pulsar Fl y seguidamente el botón de datos técnicos. Anotaremos el nombre de la tabla de base de datos y del campo.

Es posible que mientras se está procesando una sesión de Batch Input, el sistema caiga provocando la pérdida de la misma. Cuando el sistema vuelva a la situación normal, la sesión aparentemente se encuentra en estado Procesando. En realidad esto no es cierto ya que la sesión no está haciendo nada, pero tampoco hemos perdido nada. El sistema habrá ejecutado todas las transacciones hasta el momento de la caída, y podemos recuperar de una manera segura el resto de la sesión de la siguiente forma:

Desde la transacción SM35, marcar la sesión de Batch Input en cuestión. Elegir Juego de datos -> Liberar . En ese momento la sesión pasa a modo a procesar y podemos ejecutar las transacciones que faltaban.

Antes de procesar una sesión de Batch Input podemos comprobar si los datos de entrada y la secuencia de pantallas que hemos programado es la esperada. Para ello desde la SM35 seleccionaremos la sesión que querernos analizar y haremos:

Pasar a → Análisis → Juego de datos.

Si se está ejecutando una transacción en modo Invisible, podemos ir viendo el "Log" de proceso de las transacciones que se van ejecutando. Una utilidad práctica es, en el caso de un elevado número de transacciones, mirar el tiempo de proceso de una transacción y extrapolar este dato para todo el proceso, para tener una idea de la hora en la que finalizará el proceso.

Antes de realizar un programa de Batch Input es aconsejable asegurarse de que SAP no disponga ya del mismo. Por ejemplo, SAP nos ofrece bastantes Batch Inputs para carga de datos. Por ejemplo:

- Carga de clientes.
- Carga de proveedores.
- Carga de documentos contables
- Carga de pedidos pendientes.
- Carga de condiciones.
- Carga de stocks ...

Nótese que entre la fase de generación y la fase de procesado, existe un tiempo indeterminado. Si este tiempo es muy grande, es posible que durante la fase de procesado se produzcan numerosos errores, ya que es posible que haya cambiado el estado en el que se llevo a cabo la fase de generación.

Por ejemplo, si generamos una sesión de Batch Input donde se intenta modificar un cierto material, y antes de que se mande procesar esta sesión, el material se da de baja, durante la ejecución de la sesión el sistema se quejará de que dicho material no existe.

Otra posible causa de errores muy común durante el procesamiento de sesiones, es que en aquellos campos que tienen tablas de verificación,

introduzcamos valores que no estén dados de alta en las tablas de verificación. Por ejemplo, si indicamos una sociedad que no está en la tabla TOO1 (sociedades).

Otra manera de lanzar sesiones de Batch Input es ejecutando el report RSBDCSUB. Por ejemplo podemos ejecutar la sesión de Batch Input inmediatamente después de ser generada, llamando a este report con los parámetros adecuados desde el mismo programa ABAP/4 que genera la sesión.

Codificación de Batch Inputs.

Hasta ahora hemos visto que la técnica del Batch Input consiste en la generación de una sesión con los datos a introducir en el sistema y el procesamiento de los datos en el sistema destino. En este apartado veremos cómo codificar el Batch Input para generar sesiones de este tipo y otras dos técnicas más de Batch Input (CALL TRANSACTION y CALL DIALOG).

Para introducir los valores en las distintas pantallas de cada transacción utilizaremos una tabla interna con una estructura estándar. (BDCDATA).

```
DATA: BEGIN OF <tab_B_I> 0CCURS <n>.
                INCLUDE STRUCTURE BDCDATA.
DATA: END OF <tab_B_I>.
```

Los campos que componen esta tabla interna son:

- PROGRAM: Nombre del programa donde se realiza el tratamiento de cada pantalla (Dynpro) de la transacción.
- DYNPRO: Número de la pantalla de la cual queremos introducir datos.
- DYNBEGIN: Indicador de que se inicia una nueva pantalla.
- FNAM: Campo de la pantalla. (35 Caracteres como máximo).
- FVAL: Valor para el campo de la pantalla. (80 Caracteres como máximo).

Obtendremos la información del nombre del programa y el nombre del dynpro con Sistema → Status.

Obtendremos el nombre del campo con Fl (Datos Técnicos) o podemos ver todos los campos de una pantalla con el screen painter (Field list).

En esta tabla interna grabaremos un registro por cada campo de pantalla que informemos y un registro adicional con la información de cada pantalla.

El primer registro de cada pantalla, en la tabla interna tab_B_I, contendrá los datos que identifican la pantalla: Nombre del programa (PROGRAM), nombre de la pantalla (DYNPRO), y un indicador de inicio de dynpro (DYNBEGIN).

Ejemplo:

Transacción: FSSI
Programa: SAPMF02H
Dynpro : 0102

```
tab_B_I-PROGRAM =”SAPMF02H”.
tab_B_I- DYNPRO =”0102”.
tab_B_I- DYNBEGIN =”X”.
APPEND tab_B_I.
```

Seguidamente para cada campo de la pantalla que informemos, grabaremos un registro rellenando únicamente los campos FNAM (con el nombre del campo de pantalla) y FVAL (con el valor que le vamos a dar).

Ejemplo:

Rellenar campo RF02H-SAKNR con variable VAR_CTA.
Rellenar campo RF02H-BUKRS con variable VAR_SOC.

```
CLEAR tab_B_I.
tab_B_I -FNAM =”RF02H-SAKNR”.
tab_B_I -FVAL = VAR_CTA.
APPEND tab_B_I.

CLEAR tab_B_I.
tab_B_I -FNAM =”RF02H-BUKRS”.
tab_B_I -FVAL = VAR_SOC.
APPEND tab_B_I.
```

El programa Batch Input tiene que formatear los datos tal y como lo haría el usuario manualmente. Teniendo en cuenta que:

- Sólo se permiten caracteres.
- Los valores han de ser de menor longitud que la longitud de los campos.
- Si los valores de entrada son de longitud menor que el campo SAP, tendremos que justificar a la izquierda.

Si necesitamos informar campos que aparecen en pantalla en forma de tabla, tendremos que utilizar índices para dar valores a cada línea de pantalla y grabar en la tabla interna un registro por cada línea de pantalla.

Ejemplo:

```
CLEAR tab_B_I.
tab_B_I -FNAM =”campo(índice)”.
tab_B_I -FVAL =”valor”.
APPEND tab_B_I.
```

Si necesitamos proveer de una tecla de función a la pantalla, usaremos el campo BDC_OKCODE. El valor del campo será el número de la tecla de función precedido de una barra inclinada.

Ejemplo:

```
CLEAR tab_B_I.
tab_B_I -FNAM = BDC_OKCODE.
tab_B_I-FVAL = “/13”.                    “F13= Grabar.
APPEND tab_B_I.
```

También utilizamos el campo BDC_OKCODE para ejecutar funciones que aparecen en la barra de menús. Para saber el código de la función, Pulsar Fl sin soltar el botón del ratón, sobre el menú deseado.

Si necesitamos colocar el cursor en un campo en particular, usaremos el campo BDC_CURSOR. El valor del campo será el nombre del campo donde nos queremos situar.

Ejemplo:

```
CLEAR tab_B_I.
tab_B_I -FNAM = BDC_CURSOR.
tab_B_I-FVAL = "RF02H-BUKRS"
APPEND tab_B_I.
```

Para insertar sesiones en la cola de Batch Input, seguiremos los siguientes pasos en la codificación:

1.- Abrir la sesión de Batch Input utilizando el modulo de función BDC_OPEN_GROUP.

2.- Para cada transacción de la sesión:

2.a.- Llenaremos la tabla tab_B_I para entrar los valores de los campos en cada pantalla de la transacción.

2.b.- Transferir la transacción a la sesión, usando el módulo de función BDC_INSERT.

3.- Cerrar la sesión usando BDC_CLOSE_GROUP.

A continuación veremos cómo funcionan los módulos de función que necesitamos para generar un Batch Input.

BDC_OPEN_GROUP: Este módulo de función nos permite abrir una sesión. En el programa no podemos abrir otra sesión hasta que no se hayan cerrado todas las sesiones que permanezcan abiertas:

```
CALL FUNCTION "BDC_OPEN_GROUP"
          EXPORTING

CLIENT          = <mandante>
GROUP           = <nombre_sesión>
HOLDDATE  = <fecha>
KEEP        = <indicador>
USER        = <usuario>
EXCEPTIONS

CLIENT_INVALID          = 01
DESTINATION_INVALID  = 02
GROUP_INVALID           = 03
HOLDDATE_INVALID     = 04
INTERNAL_ERROR       = 05
QUEUE_ERROR             = 06
RUNNING                 = 07.
```

Donde:

- CLIENT: Es el mandante sobre el cual se ejecutará la sesión de Batch Input, si no se indica este parámetro se tomará el mandante donde se ejecute el programa de generación de la sesión.

- GROUP: Nombre de la sesión de Batch Input, con la que identificaremos el juego de datos en la transacción SM35 de tratamiento de Batch Input.

- HOLDDATE : Si indicamos este parámetro, el sistema no permitirá ejecutar la sesión hasta que no sea la fecha indicada. Sólo el administrador del sistema podrá ejecutar una sesión antes de esta fecha.

- KEEP: Si informamos este parámetro con una "X", la sesión será retenida en el sistema después de ser ejecutada y sólo un usuario con autorizaciones apropiadas podrá borrarla.

- USER: Es el usuario que de ejecución de la sesión.

BDC_INSERT: Este módulo de función inserta una transacción en la sesión de Batch Input:

```
CALL FUNCTION "BDC-INSERT"
          EXPORTING

                    TCODE          = <Transacción>
          TABLES

                    DYNPROTAB = <Inttab>
          EXCEPTIONS

                    INTERNAL_ERROR = 01
                    NOT_OPEN            = 02
                    QUEUE_ERROR         = 03
                    TCODE_INVALID       = 04.
```

Donde:

- TCODE: Es el código de la transacción que vamos a simular.
- DYNPROTAB: Es la tabla interna, con estructura BDCDATA, donde especificamos la secuencia de pantallas de la transacción y los distintos valores que van a tomar cada campo que aparece en cada pantalla.

BDC_CLOSE_GROUP: Con esta función cerraremos la sesión una vez ya hemos transferido todos las datos de las transacciones a ejecutar:

```
CALL FUNCTION "BDC_CLOSE_GROUP"
     EXCEPTIONS
          NOT_OPEN            = 01
          QUEUE_ERROR         = 02.
```

Podemos resumir las características de la técnica de las sesiones de Batch Input:

- Procesamiento retardado (asíncrono).
- Transferencia de datos a múltiples transacciones.
- Actualización de la base de datos inmediata (síncrona). No se ejecuta una nueva transacción hasta que la anterior no actualiza los datos en base de datos.
- Generación de un "Log" para cada sesión.

- Imposibilidad de generar varias sesiones simultáneamente desde un mismo programa.

Como ya hemos citado anteriormente existen otras dos técnicas de Batch Input, el CALL TRANSACTION y el CALL DIALOG. En ambas, a diferencia de la técnica de sesiones, la ejecución de las transacciones es inmediata, es decir la ejecución de las transacciones es controlada por nuestro programa ABAP/4 y no posteriormente desde la SM35 lo cual puede resultar interesante en ciertas ocasiones.

CALL TRANSACTION:

Características:

- Procesamiento síncrono.
- Transferencia de los datos a una única transacción.
- Actualización de la base de datos síncrona y asíncrona. El programa decide qué tipo de actualización se realizará.
- La transacción y el programa que la llama tendrán áreas de trabajo (LUW) diferentes. El sistema realiza un COMMIT WORK inmediatamente después del CALL TRANSACTION.
- No se genera ningún "Log" de ejecución.

- Como se utilizará la técnica del CALL TRANSACTION:

En primer lugar llenaremos la tabla BDCDATA de la misma manera que hemos explicado a lo largo de este capítulo.

Usar la instrucción CALL TRANSACTION para llamar a la transacción:

```
CALL TRANSACTION <transacción>
          USING     <tabint>
          MODE <modo_ejec>
          UPDATE    <tipo_actual>.
```

Donde:

- <tabint> Tabla interna (con estructura BDCDATA)

- <modo_ejec> Modo ejecución.

Puede ser: "A" Ejecución visible.
"N" Ejecución invisible. Si ocurre algún error en la ejecución de la transacción el código de retorno será distinto de cero.
"E" Ejecución visualizando sólo errores.

- <tipo_actual> Tipo de actualización en la base de datos.
  Puede ser : "S" Actualización Síncrona. (Inmediata).
  "A" Actualización Asíncrona. (Hasta que no termina la transacción no graba en la BDD).

Después del CALL TRANSACTION podemos comprobar si SY-SUBRC es 0, en cuyo caso la transacción se habrá ejecutado correctamente. En caso contrario, SAP llena otros campos del sistema que contienen el número, identificación, tipo y variables del mensaje online que haya emitido la transacción en el momento del error.

SY-MSGID = Identificador de mensaje.
SY-MSGTY = Tipo de mensaje (A,E,I,W...)
SY-MSGNO = Número de mensaje.
SY-MSGV1... SY-MSGV4 = Variables del mensaje.

De modo que para ver que ha ocurrido podemos ejecutar la instrucción:

```
MESSAGE ID SY-MSGID
     TYPE SY-MSGTY NUMBER SY-MSGNO
WITH SY-MSGVI SY-MSGV2 SY-MSGV3 SY-MSGV4.
```

CALL DIALOG:

Características:

- Procesamiento síncrono.
- Transferencia de los datos a una única transacción.

- La transacción y el programa tendrán la misma área de trabajo (LUW). Es decir hasta que en el programa no se realiza un COMMIT WORK no se actualiza la base de datos.

- No se genera ningún "Log" de ejecución.

Para utilizar la técnica del CALL DIALOG, llenaremos la tabla BDCDATA para usar la instrucción CALL DIALOG para llamar a la transacción:

```
CALL DIALOG <Mod_diálogo>
     USING     <tabint>
     MODE <modo_ejec>.
```