

# **CICS**

(CUSTOMER INFORMATION CONTROL SYSTEM)

---

Marzo de 1999

Material para el participante

## CUSTOMER INFORMATION CONTROL SYSTEM

El CICS nos permite trabajar en forma ON-LINE o interactiva con programas que hasta ahora hemos visto trabajar en forma BATCH, con lo cual nos produce un carga inicial de información más lenta pero un procesamiento posterior mucho más rápida.

Como comienza a ejecutarse un programa CICS?

Simplemente colocando el nombre de la transacción que “arranca” el programa COBOL/CICS. En este caso no necesitamos, para trabajar en forma ON-LINE con un programa CICS, de un lenguaje de control de tareas (JCL). Todo esta definido y relacionado en unas tablas que se linkeditan junto con el CICS en el momento del arrancar a este.

Luego de arrancada una transacción, esta ejecuta un programa COBOL que contiene instrucciones CICS. Cada instrucción CICS internamente ejecuta un CALL produciendo una interrupción de entrada/salida que puede ser atendida por el sistema operativo (lectura o grabación de un registro, llamada a otro programa, etc.) o por un operador (completar los datos de una pantalla, confirmar una respuesta, etc.). Las interrupciones que requieren la intervención de un operador se satisfacen con en ingreso de algún dato y tocando algunas de las teclas de función que son atendidas a controladas por el programa.

Transacciones: Pueden tener de 1 a 4 caracteres alfanumérico sin caracteres especiales y debe respetar las palabras reservadas para el CICS.

Formato de los comandos

EXEC CICS función [opciones [(argumento)]]....

Ej.

```
EXEC CICS READ FILE ('FILEA')  
          RIDFLD(KEYVAL)  
          INTO(AREA_FILEA)  
          END-EXEC.
```

### Definición de tablas de recursos.

Todo el CICS podría resumirse en las cuatro tablas que a continuación describiremos, las cuales cada una cumple una función muy específica.

<b>TCT</b>  TERMINAL CONTROL TABLE  TERMINAL DEFINITIONS	<b>PCT</b>  PROGRAM CONTROL TABLE  TRANSACTION DEFINITIONS
<b>PPT</b>  PROCESSING PROGRAM TABLE  PROGRAM DEFINITIONS	<b>FCT</b>  FILE CONTROL TABLE  FILE DEFINITIONS

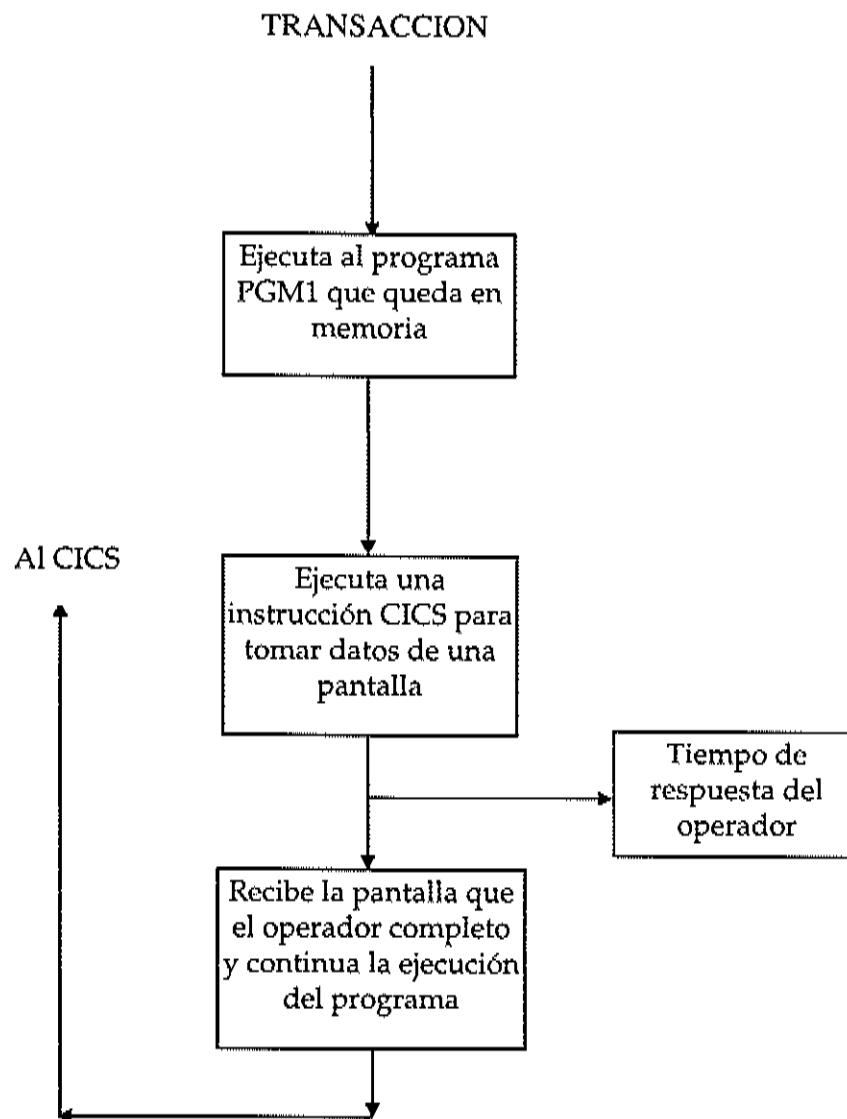
- TCT: Tabla de control de terminales. En esta tabla se ingresan y definen todas las terminales que se habilitarán para que se puedan ejecutar todos los programas CICS. Cuando se habla de terminales son pantallas e impresoras que están afectadas al sistema que se define.
- PCT: Tabla de control de programas. Se definen todas las transacciones que se pueden ejecutar en cada una de las terminales definidas en la TCT. No todas las transacciones se pueden ejecutar en todas las terminales.
- PPT: Definición de programas. Aquí se indica cual es la relación que existe entre el programa y la transacción. Es decir, que transacción arranca la ejecución de que programa.
- FCT: Definición de archivos. Aquí se definen las relaciones de los nombres de archivos descriptos en CICS con los nombres reales de cada archivo físico.

La definición, actualización y mantenimiento de todas estas tablas queda a cargo del System Programmer de la instalación o el encargado del CICS.

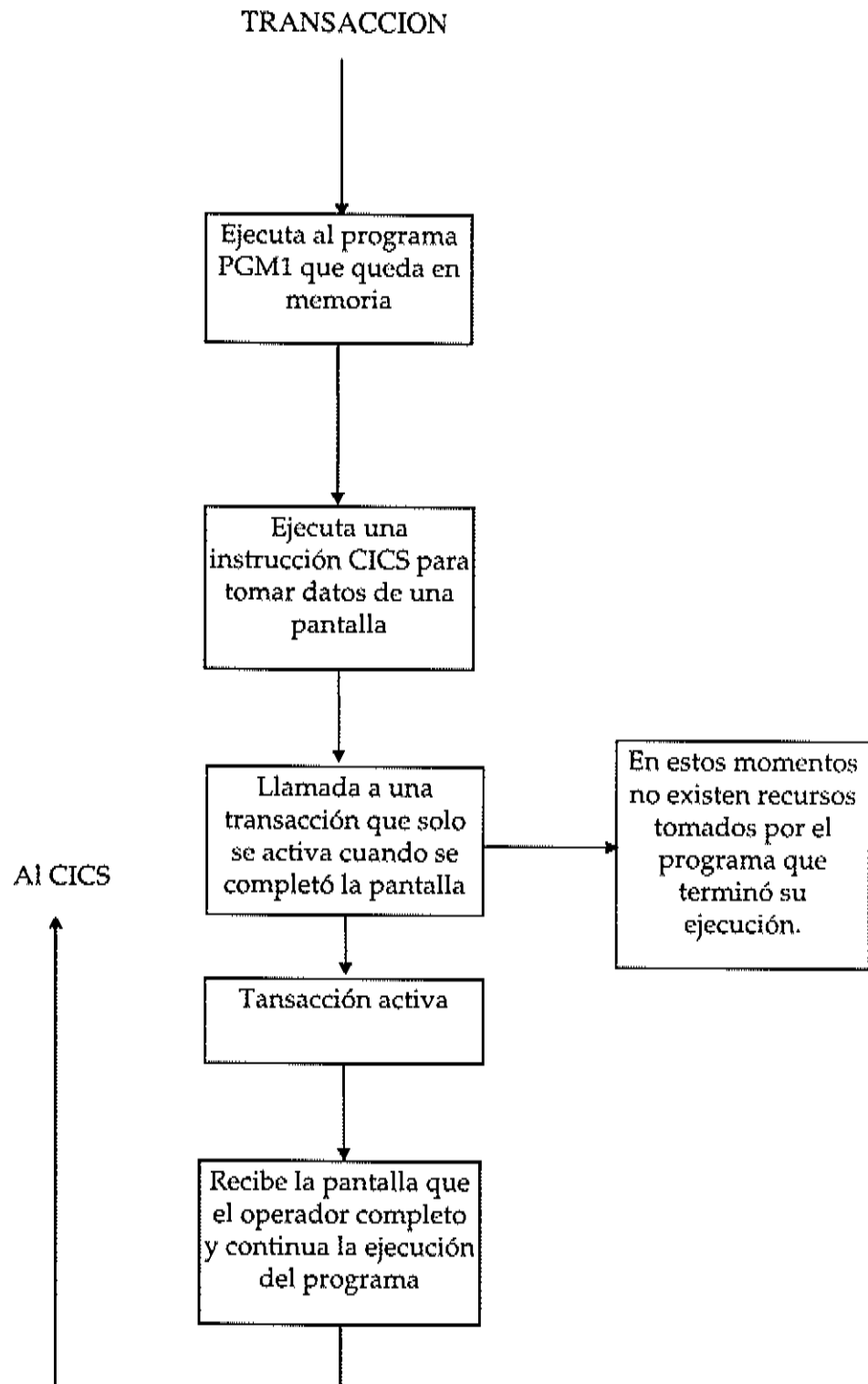
## FORMAS DE PROGRAMACION

Se dividen en dos grupos.

- **CONVERSACIONALES.** Son los que menos se utilizan en cuando a la forma de programar. Cada transacción comienza la ejecución de un programa en tiempo de máquina, con cada ejecución de una instrucción EXEC CICS ..... END-EXEC se produce una interrupción de entrada/salida, además si una de estas instrucciones necesita la respuesta de un operador, no debemos olvidarnos que las máquinas procesan la información en nanosegundos y por mas que tengamos a un operador muy entrenado, puede tardar varios segundos o minutos en contestarle o completar los datos de una pantalla, con lo cual el programa queda cargado en memoria durante este proceso.



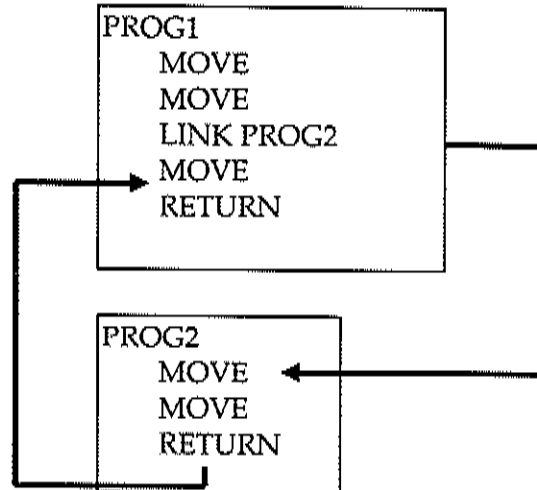
- **PSEUDOCONVERSACIONALES.** Es el tipo de programación que más frecuentemente se utiliza, porque por cada interrupción de entrada/salida que requiere la intervención de un operador, en lugar que el programa se quede en memoria, el programa llama a otra transacción que queda activa hasta que el operador lo decide, pero el programa liberó todos los recursos para que así la máquina pueda seguir ejecutando otros programas durante el tiempo en que el operador efectúa su trabajo.



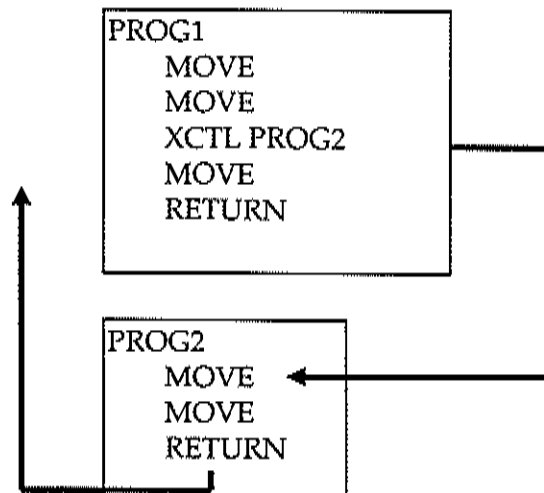
## Control de programas

- LINK
- XCTL
- LOAD

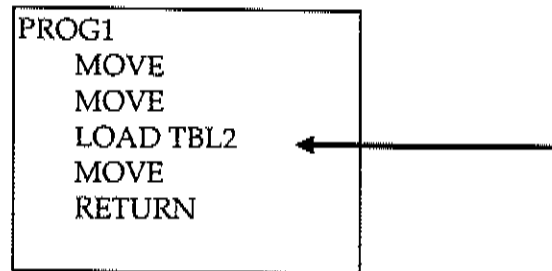
### LINK



### XCTL

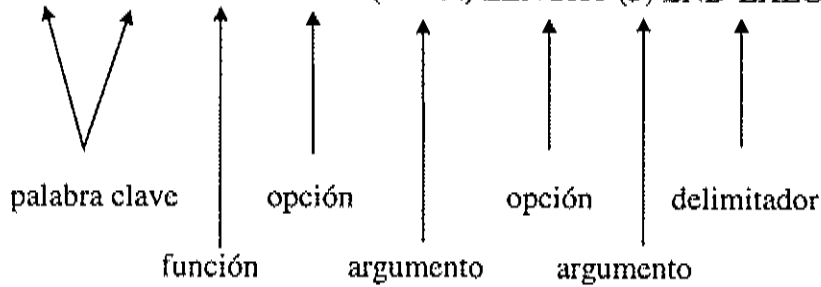


## LOAD



A continuación se describirá la sintaxis de alguna instrucciones CICS como referencia.

EXEC CICS SEND FROM (FLDA) LENGHT (5) END-EXEC.



WORKING-STORAGE SECTION.

```

01  WRLFLDS.
    05  INL                      PIC S9(4) COMP.
01  INAREA.
    05  TRANID                   PIC X(4).
    05  FILLER                   PIC X.
    05  IMSG.
        10  TERMCODE             PIC X(4).
        10  FILLER               PIC X(18).
  
```

PROCEDURE DIVISION.

```

    MOVE 9 TO INL.
    EXEC CICS RECEIVE INTO (INAREA)
      LENGHT (INL) END-EXEC.
  
```

Aquí se muestra como se puede recibir algún dato desde una terminal del tipo 3270, en que será colocado en un área definido en la WORKING-STORAGE llamado INAREA cuya longitud máxima preparada para aceptar es de 9 bytes.

También se definen condiciones de excepción como en el ejemplo que sigue.

```
.....
WORKING-STORAGE SECTION.
01  WRLFLDS.
    05  INL                      PIC S9(4) COMP.
    05  RCVERR                   PIC S9(8) COMP.
01  INAREA.
    05  TRANID                   PIC X(4).
    05  FILLER                   PIC X.
    05  IMSG.
        10  TERMCODE            PIC X(4).
        10  FILLER              PIC X(18).
.....
PROCEDURE DIVISION.
    MOVE 9 TO INL.
    EXEC CICS RECEIVE INTO (INAREA) RESP (RCVERR)
        LENGHT (INL) END-EXEC.
    IF RCVERR EQUAL TO DFHRESP(LENGERR) THEN
        MOVE 'DATO DE ENTRADA DE DISTINTA LONGITUD' TO MSG1
.....
```

Aquí se establece una condición de excepción cuando la longitud del dato recibido es diferente de la esperada.

Entre diferentes tareas (programas) se pueden pasar datos, como se muestra a continuación.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. LAMODC1A.
WORKING-STORAGE SECTION.
01  COMSTART.
    05  STATUS-I.
        10  STARTED-I          PIC X VALUE 'N'.
        10  ENDED-I            PIC X VALUE 'N'.
    05  RATE-I                 PIC X(78).
.....
PROCEDURE DIVISION.
.....
*      MOVE INFORMATION TO RATE-I.
.....
    EXEC CICS RETURN TRANSID ('LAC2') COMMAREA (COMSTART)
        LENGHT(80) END-EXEC.
    GOBACK.
.....
```



PROGRAM-ID. LAMODC1B.  
WORKING-STORAGE SECTION.

01 .....  
LINKAGE SECTION.  
01 DFHCOMMAREA.  
    05 STATUS-C.  
        10 STARTED        PIC X.  
        10 ENDED          PIC X.  
    05 RATE                PIC X(78).

.....  
PROCEDURE DIVISION.

.....  
\*ACA SE ACCEDE A LOS CAMPOS DEFINIDOS EN LA DFHCOMMAREA.  
.....

GOBACK.

COMSTART. Es el comienzo del área de trabajo de la working que contiene los datos a transferir del programa LAMODC1A asociado a la tarea LAC1.

La instrucción RETURN transfiere el control a la tarea LAC2 a través del TRANSID, además la dirección de memoria en la COMMAREA y la longitud con LENGHT de esta.

La transacción LAC2 recibe la dirección del campo COMSTART que tiene una longitud de 80 bytes.

Para mostrar datos en la pantalla de una terminal del tipo 3270, se deben definir mapas que contienen los distintos datos a presentar y recibir de un operador. Estos mapas que en CICS son manejados por BMS (Basic Mapping Support), son utilizados en la WORKING de un programa COBOL y sirven para transferir datos a la pantalla y recibir los resultados de la misma.

La confección de un mapa se debe entender como un gran registro de hasta 1920 Bytes de longitud (24 x 80 cantidad de caracteres de la terminal de tipo 3270).

Tiene tres tipos de datos a definir.

- DFHMSD: Indicaciones generales del mapset.
- DFHMDI: Tamaño del mapa.
- DFHMDF: Definición de cada campo del mapset.

Los mapas se linkeditan y el resultado es un mapa físico en una librería y un copy en otra librería para utilizarlo en la WORKING para poder referirse a los campos de entrada o de salida del mapa.

Dentro de este copy, CICS antepone 12 bytes al comienzo del mapa y tres bytes por cada campo definido para poder colocar información acerca de cada campo. Esta información es la longitud del campo y atributos que se pueden modificar o consultar desde el programa.

La instrucción para enviar un mapa a la terminal es la siguiente.

```
EXEC CICS SEND MAP('ORDERM')  
          MAPSET('MAPSETA')  
          END-EXEC.
```

ORDERM: Nombre generado en el copy y que se encuentra en la WORKING del programa.

MAPSETA: Nombre del mapa físico definido en las tablas de CICS (instrucción DFHMSD).

Además, cada instrucción CICS contiene distintas opciones que se verán en el curso y nuevas instrucciones para manejo de archivos.