

¿CÓMO INTERPRETAR LOS CICLOS EN COBOL CICS?

OBJETIVO

- Los ciclos en COBOL dentro de un entorno CICS no se manejan como los típicos bucles **PERFORM** que se arman en COBOL puro.
- En CICS, el ciclo más importante es el **flujo de ejecución de una transacción**, que involucra varios pasos desde que el usuario interactúa con una pantalla hasta que se devuelve una respuesta.
- Cada **CICLO** representa una **UNIDAD DE TRABAJO** en CICS
- En este documento; vamos a desglosarlo de forma clara y estructurada

ESPECIFICACIONES



Ciclo (Unidad de trabajo) típico de una transacción COBOL-CICS

Este ciclo representa cómo se ejecuta una transacción en CICS usando un programa COBOL; lo desarrollaremos paso a paso:

1. Interacción con el usuario (pantalla/mapa)

- El usuario ingresa datos en una pantalla diseñada con **BMS (Basic Mapping Support)**.
- El mapa se define en assembler o con herramientas específicas.

2. Recepción de datos (**EXEC CICS RECEIVE MAP**)

- El programa COBOL usa esta sentencia para recibir los datos del mapa.

```
EXEC CICS  
  RECEIVE MAP (MAPCLI1) MAPSET (MAPSET1)  
END-EXEC.
```

3. Validación y lógica de negocio

- Se analiza lo ingresado, se validan campos, y se decide qué hacer (consultar en base de datos DB2 o archivo VSAM, mostrar error, etc.).

- Aquí puede haber un bucle COBOL tal como:

```
PERFORM UNTIL END-OF-FILE

EXEC CICS
  READ DATASET ('PERSONA')
  RIDFLD (WS-USER-DATA)
  INTO (REG-PERSONA)
  LENGTH (CT-DATASET-LEN)
  EQUAL
  RESP (WS-RESP)
END-EXEC

MOVE datos TO campos-de-salida

END-PERFORM.
```

4. Envío de datos a la pantalla (EXEC CICS SEND MAP)

- Se devuelven los datos al usuario en el mapa:

```
EXEC CICS
  SEND (MAPCLI1) MAPSET (MAPSET1)
  FROM(WS-DATOS) ERASE
END-EXEC.
```

5. FINAL CICLO o UNIDAD DE TRABAJO - (EXEC CICS RETURN TRANSID)

- El programa termina su ejecución (liberando recursos tomados) y espera la próxima acción del usuario (es lo que llamamos PSEUDO-CONVERSACIONAL):

```
EXEC CICS
  RETURN TRANSID('BD1F')
  COMMAREA (WS-COMMAREA)
END-EXEC.
```

¿Dónde aparecen los ciclos COBOL?

Deberemos diferenciar un **bucle** de un **ciclo** en COBOL CICS.

Los bucles COBOL como **PERFORM UNTIL**, **PERFORM VARYING**, etc., se usan para:

- Recorrer resultados de una consulta SQL.
 - Procesar múltiples registros.
 - Validar múltiples campos.
 - Generar líneas en una tabla de salida.
-

Los ciclos o **UNIDADES DE TRABAJO en COBOL CICS**; Se resuelven en un proceso completo de una transacción; por ejemplo: **consulta clientes** y muestra los resultados en una pantalla.

Este ejemplo incluye:

- Un mapa llamado **MAPCLI1**
 - Acceso a datos en archivo VSAM KSDS
 - Uso de bucles COBOL
 - Flujo CICS completo; por CICLO o UNIDADES DE TRABAJO
-

Estructura general del programa

IDENTIFICATION DIVISION.
PROGRAM-ID. PGMPRD1F.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.

DATA DIVISION.
WORKING-STORAGE SECTION.

* Variables para mapa
COPY MAP1D1F.

* Variables generales
01 CT-DATASET PIC X(08) VALUE 'PERSONA'.

01 CT-DATASET-LEN PIC S9(04) COMP VALUE 160.

PROCEDURE DIVISION.

MAIN-PROCESS.

LUEGO DE ESTABLECER CICLO > 0 - LA DFHCOMMAREA CONTIENE DATOS

```
EXEC CICS
  RECEIVE MAP(MAPCLI1) MAPSET(MAPSET1)
END-EXEC.
```

```
MOVE TIPDO1I TO WS-USER-TIPDOC
MOVE NUMDO1I TO WS-USER-NRODOC
```

```
MOVE WS-USER-TIPDOC     TO TIPDO1O
MOVE WS-USER-NRODOC     TO NUMDO1O
```

```
EXEC CICS
  READ DATASET ('PERSONA')
    RIDFLD (WS-USER-DATA)
    INTO (REG-PERSONA)
    LENGTH (CT-DATASET-LEN)
    EQUAL
    RESP (WS-RESP)
```

```
END-EXEC
```

EVALUATE WS-RESP

```
WHEN DFHRESP(NORMAL)
  MOVE WS-USER-TIPDOC     TO TIPDO1O
  MOVE WS-USER-NRODOC     TO NUMDO1O
  MOVE PER-CLI-NRO        TO NROCLIO
  MOVE PER-NOMAPE        TO NOMAPEO
  MOVE PER-DIRECCION      TO DIREO
  MOVE PER-TELEFONO       TO TELEO
  MOVE PER-EMAIL         TO EMAILO
  MOVE CT-MNS-06         TO MSG1O
  MOVE LENGTH OF MAP1D1FO TO WS-LONG
```

```
WHEN DFHRESP(NOTFND)
  MOVE WS-USER-TIPDOC     TO TIPDO1O
  MOVE WS-USER-NRODOC     TO NUMDO1O
  MOVE CT-MNS-03         TO MSG1O
```

```
WHEN OTHER
```

```
MOVE WS-USER-TIPDOC    TO TIPDO10
MOVE WS-USER-NRODOC    TO NUMDO10
MOVE CT-MNS-08         TO MSG10
```

```
END-EVALUATE.
```

```
EXEC CICS
  SEND MAP ('MAPCLI1') MAPSET ('MAPSET1')
  FROM (MAP-DATA) ERASE
END-EXEC.
```

```
EXEC CICS
  RETURN TRANSID ('AD1F')
  COMMAREA (WS-COMMAREA)
END-EXEC.
```

 Mapa **MAPCLI1** (simplificado)

Este mapa tendría campos como:

- **ID-CLIENTE** para ingresar el ID
- **NOMBRE y DIRECCION** para mostrar resultados
- **MENSAJE** para mostrar acciones a seguir por parte del usuario o errores detectados

 ¿Qué conceptos se integran aquí?

Componente	Función
RECEIVE MAP	Captura datos ingresados por el usuario
SQL DECLARE/FETCH	Consulta DB2 usando cursor para múltiples filas como resultado de la ejecución de una query
PERFORM UNTIL	Bucle COBOL para recorrer resultados
SEND MAP	Devuelve datos al usuario en pantalla
RETURN TRANSID	Finaliza el ciclo y espera nueva acción

Todos estos conceptos forman parte de un mismo CICLO en COBOL CICS.









EXPRESIÓN 'CICLO 0' EN CICS (PRIMERA UNIDAD DE TRABAJO en una transacción)

- Se refiere al primer ingreso del usuario a una transacción, antes de que haya interacción o datos disponibles recibidos a través de **LINKAGE SECTION**.
- Es un concepto práctico que se usa para diferenciar el estado inicial de una transacción de los ciclos o unidades de trabajo posteriores, donde ya hay datos ingresados a través de **LINKAGE SECTION**.



Diferencias entre el ciclo 0 y los demás ciclos en CICS

Acción	Ciclo 0 (Inicial)	Ciclos posteriores (Iterativos)
 Interacción del usuario	No hay datos ingresados aún	El usuario ya ha ingresado datos en el mapa
 RECEIVE MAP	Puede no ejecutarse, o se usa para inicializar campos a través de DFHRESP = MAPFAIL	Se usa para capturar datos ingresados
 Lógica de negocio	Se muestra pantalla vacía o con valores por defecto	Se ejecuta lógica según los datos ingresados
 Acceso a datos almacenados en disco	No se realiza consulta aún	Se consulta la base de datos o archivo VSAM con los datos del usuario
 SEND MAP	Se envía mapa vacío o con mensaje de bienvenida	Se envía mapa con resultados, errores o confirmaciones
 Flujo de control	Se prepara el entorno para la interacción	Se procesa y se puede repetir según la lógica del negocio; hasta que el usuario indique el FIN a través de la tecla de función correspondiente



¿Por qué es importante distinguir el ciclo 0?

- Permite inicializar variables y estructuras antes de procesar datos.
 - Evita errores por campos vacíos o no definidos.
 - Mejora la experiencia del usuario al mostrar una pantalla limpia o con datos.
 - Facilita el diseño de mapas y programas más robustos.
-



Ejemplo de detección del ciclo 0 en COBOL-CICS

```
MOVE DFHCOMMAREA TO WS-COMMAREA.

IF EIBCALEN = 0
    MOVE LENGTH OF MAPCLI10 TO WS-LONG
    MOVE LOW-VALUES TO MAPCLI10
    MOVE 'Ingrese la opción deseada:' TO MENSAJE

    EXEC CICS
        SEND MAP (MAPCLI1)    MAPSET (MAPSET1)
        FROM (MAPCLI10) ERASE
    END-EXEC

    EXEC CICS
        RETURN
        TRANSID ('BD1F')
        COMMAREA (WS-COMMAREA)
    END-EXEC

ELSE

    EXEC CICS
        RECEIVE MAP (MAPCLI1)
        MAPSET (MAPSET1)
        INTO (MAPCLI11)
        RESP(WS-RESP)
    END-EXEC

END-IF.
```

¿Cómo se interpreta esta porción de código?

- **EIBCALEN = 0** Verifica el largo de la **DFHCOMMAREA**
- Si se cumple la condición; **estamos en el ciclo 0 o primera unidad de trabajo**: no hay datos ingresados aún.
 - Se envía el mapa con un mensaje con instrucción al usuario, y se retorna el control a CICS incluyendo la **DFHCOMMAREA**.

```
EXEC CICS
  RETURN
  TRANSID ('AD1F')
  COMMAREA (WS-COMMAREA)
END-EXEC.
```

De esta manera resulta PSEUDO-CONVERSACIONAL; libera los recursos tomados.

- Si en cambio NO cumple la condición; **estamos en cualquier otro ciclo > 0**: ya hubo interacción con el usuario; entonces se reciben los datos a través de la **DFHCOMMAREA** y se arman los caminos funcionales que responden a la solución.

- Finalizando con:

```
EXEC CICS
  SEND MAP (MAPCLI1) MAPSET (MAPSET1)
  FROM (MAPCLI1O) ERASE
END-EXEC

EXEC CICS
```



```
RETURN
TRANSID ('BD1F')
COMMAREA (WS-COMMAREA)
```

```
END-EXEC.
```

Excepto que el usuario decida salir de la aplicación (mediante tecla de función específica: **PF12**) y en ese caso se construye el siguiente camino lógico de salida final; **DEVUELVE EL CONTROL AL CICS**; finalizaron **TODAS** las **UNIDADES DE TRABAJO** (a pedido del usuario) para esta transacción:

En **WORKING STORAGE SECTION**.

```
01 CT-MNS-EXIT    PIC X(72) VALUE      'FIN TRANSACCION BD1F'.
```

En **PROCEDURE DIVISION**.

```
EXEC CICS
  SEND CONTROL ERASE
END-EXEC
```

```
EXEC CICS
  SEND TEXT
    FROM (CT-MNS-EXIT)
END-EXEC
```

```
EXEC CICS
  RETURN
END-EXEC.
```