

## CICS - CONCEPTOS RELEVANTES

### 1. EIB – Exec Interface Block

- **DFHEIBLK**: estructura automática incorporada en la **LINKAGE SECTION** durante la traducción CICS.
  - **Finalidad**: Proveer a nuestro programa de información contextual sobre la ejecución: terminal, usuario, hora, resultado de comandos, etc.
  - **Campos clave**: **EIBAID**, **EIBCALEN**, **EIBDATE**, **EIBTIME**, **EIBRCODE**, **EIBTRNID**, entre otros. Se pueden consultar pero no modificar.
- 

### 2. Testeo de Ejecución de Comandos

CICS provee mecanismos para validar si los comandos fueron exitosos:

#### A. **HANDLE CONDITION**

- Maneja condiciones de error mediante bifurcaciones GO TO.
- Puede manejar hasta 16 condiciones.
- Usado generalmente al inicio del programa para controlar errores comunes y excepciones.

#### B. **RESP**

- Permite capturar el código de resultado en una variable, que se evalúa luego con **DFHRESP**.
- Es más programático que **HANDLE**.

#### C. **Combinación de ambos**

- Puede usarse **HANDLE** para condiciones generales y **RESP** para tratamiento más fino en ciertas secciones.
- 

### 3. **HANDLE AID**

- Similar a **HANDLE CONDITION** pero enfocado en teclas presionadas.
  - Se activa tras un **RECEIVE**, detecta claves como **ENTER**, **PF1–PF24**, **CLEAR**, etc.
  - Alternativa eficiente: evaluar **EIBAID** para decidir el flujo según la tecla.
- 

### 4. Manejo de Fecha y Hora

- **ASKTIME**: obtiene fecha y hora en formato binario absoluto.

- **FORMATIME**: convierte esa info a formatos legibles como **YYMMDD**, **DDMMYY**, hora **hh:mm:ss**, día de la semana, etc.
  - Muy útil para trazabilidad en registros.
- 

## 5. Manejo de Mapas

### A. **SEND MAP**

- Envía un mapa a la terminal.
- Opciones como **DATAONLY**, **MAPONLY**, **ERASE**, **CURSOR**, **FREEKB**, entre otras.
- Se requiere inicializar el área simbólica con **LOW-VALUES**.

### B. **RECEIVE MAP**

- Recibe datos desde la terminal.
  - Los formatea según la definición del mapa y los ubica en la work-area.
- 

## 6. Tratamiento de Archivos VSAM

### A. **READ**

- Lectura con opciones como **RIDFLD**, **KEYLENGTH**, **GENERIC**, **GTEQ**, **EQUAL**.
- Excepciones manejables: **NOTFND**, **LENGERR**, **DUPREC**, **IOERR**.

### B. **WRITE**

- Graba nuevos registros. Puede usar **MASSINSERT** para múltiples ingresos secuenciales.

### C. **REWRITE**

- Actualiza registros previamente leídos con **READ UPDATE**.

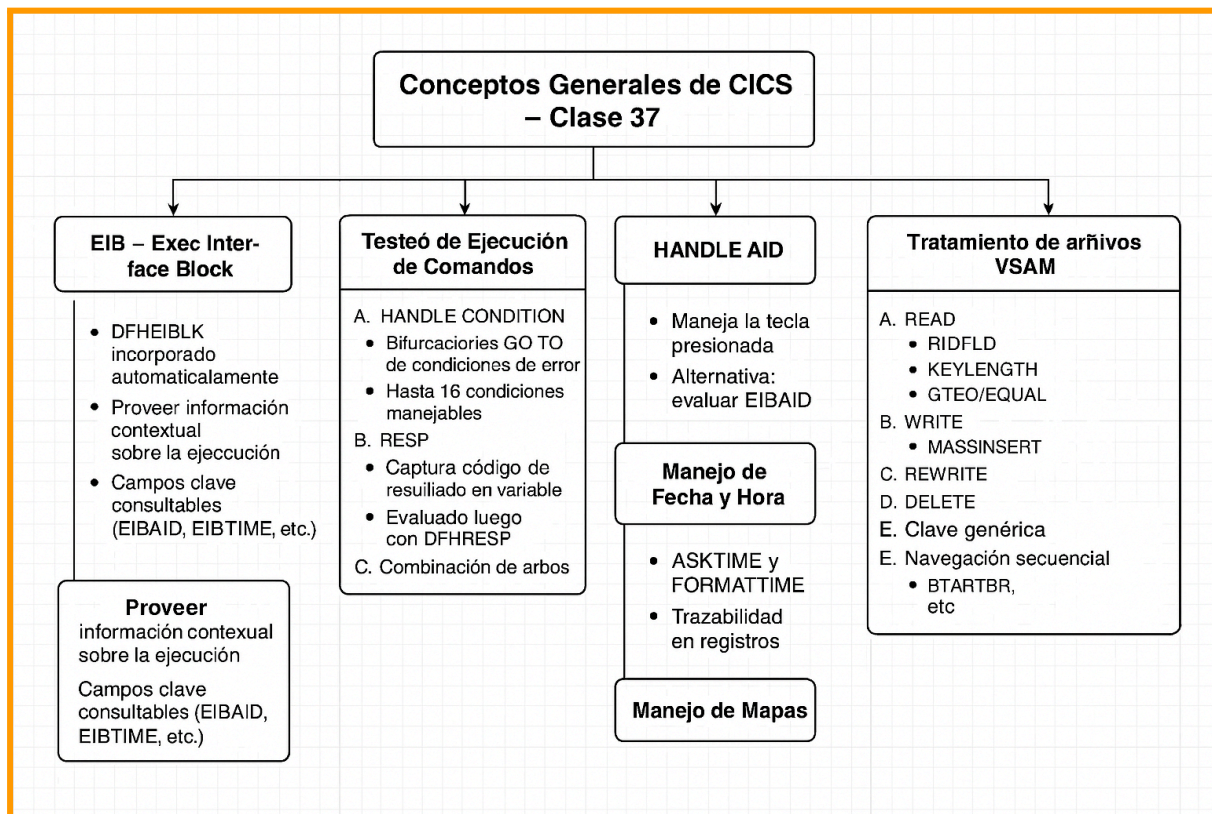
### D. **DELETE**

- Borra registros. Puede usar clave genérica y **NUMREC**.

### E. Navegación secuencial

- **STARTBR**: Posiciona para búsqueda.
  - **READNEXT**, **READPREV**: Lectura secuencial.
  - **RESETBR**, **ENDBR**: Control de navegación.
-

Todo esto forma la base de cómo interactuar con CICS para manejar terminales, datos, excepciones y archivos:



A continuación se explica cada uno de los conceptos que aparecen en el esquema visual anterior.

Este diagrama está dividido por bloques funcionales de una aplicación típica bajo CICS, donde cada elemento cumple un rol dentro del flujo de ejecución.

Explicando las partes:

## 1. EIB (Exec Interface Block)

- Es una estructura que CICS inyecta automáticamente en tu programa.
- Contiene información crítica sobre la ejecución, como el ID de transacción (**EIBTRNID**), terminal (**EIBTRMID**), hora (**EIBTIME**), estado del último comando (**EIBRCODE**), etc.
- **Tu programa puede consultarla, pero no modificarla.**

## 2. Testeo del Resultado de Comandos CICS

Después de ejecutar cualquier comando CICS, debés verificar su resultado. Tenés tres formas:

#### ✓ A. **HANDLE CONDITION**

- Intercepta condiciones de error predefinidas.
- Redirige el flujo del programa (como un **GO TO**) a un párrafo cuando se cumple esa condición.



#### B. **RESP**

- Guarda el código de resultado de un comando en una variable.
- Luego se evalúa con **EVALUATE** y **DFHRESP()**.



#### C. **Combinación**

- Podés usar ambos según el nivel de control que necesites: **HANDLE** para errores comunes, **RESP** para decisiones más específicas.



### 3. **HANDLE AID y EIBAID**

Controlan la lógica según la tecla que pulsa el usuario:

- **HANDLE AID** permite asociar teclas como **PF2**, **PF6**, **ENTER** a párrafos específicos del programa.
- **EIBAID** contiene la última tecla presionada (te da más flexibilidad y mejor performance).
- Usás uno u otro según tu estrategia de control de flujo.



### 4. **Fecha y Hora – ASKTIME y FORMATTIME**

- **ASKTIME** obtiene el tiempo absoluto desde 1900.
- **FORMATTIME** convierte ese valor a formatos legibles: fecha (**DD/MM/AA**, **YYMMDD**), hora (**hh:mm:ss**), día del mes, día de la semana, etc.
- Muy útil si necesitás trazabilidad o estampar marcas de tiempo.



### 5. **Manejo de Mapas**



#### A. **SEND MAP**

- Envía datos a la terminal en forma formateada.
- Opciones como **DATAONLY**, **ERASE**, **CURSOR** controlan qué se muestra y cómo.



#### B. **RECEIVE MAP**

- Recibe datos desde la terminal.
- Convierte la entrada en campos útiles según el mapa simbólico.

Ambos requieren haber definido los mapas previamente con BMS y mantener limpias las áreas de datos (**LOW-VALUES**).

---

## 6. Archivos VSAM

Una parte clave del flujo transaccional. Los comandos más importantes:

### **READ:**

- Lee registros del archivo.
- Puede usar claves completas o genéricas (**KEYLENGTH**, **GENERIC**).
- Opciones como **GTEQ** o **EQUAL** definen cómo buscar.

### **WRITE:**

- Inserta nuevos registros.
- Puede usarse con **MASSINSERT** para inserciones múltiples.

### **REWRITE:**

- Modifica registros que antes fueron leídos con **READ UPDATE**.

### **DELETE:**

- Elimina registros.
- Podés usar clave directa o genérica.

### **Navegación Secuencial:**

- **STARTBR**: posiciona el cursor para leer registros.
  - **READNEXT** / **READPREV**: lectura secuencial hacia adelante o atrás.
  - **RESETBR**: reinicia la búsqueda.
  - **ENDBR**: termina la búsqueda.
- 

Todo esto se orquesta para que las aplicaciones CICS sean consistentes y eficientes al interactuar con el usuario y los datos.