ESTANDARES DESARROLLO SOLUCIONES EN COBOL-CICS

Contenido

[MAPAS 2](#_Toc176891375)

[PROGRAMAS 4](#_Toc176891376)

[ELEMENTOS DB2 6](#_Toc176891377)

[AMBIENTE CICS 6](#_Toc176891378)

[CONTROL DE VERSIONES 8](#_Toc176891379)

Este documento compila los estándares utilizados para el desarrollo de soluciones en Cobol con DB2 y CICS.

## MAPAS

* Ubicación de los mapas: P390.CICS.MAPS
* Nombre del MAPSET: M*YYXX*99
  + Longitud: 7 caracteres
  + *YY*-Identificador de la aplicación, ejemplo SN (Sistema de Notas)
  + *XX*-Descriptivo de la función del mapa, ejemplo ME menú, AC- actualización o un consecutivo 99
* Nombre del Mapa: igual que el nombre del MAPSET, cuando se tienen varios mapas para el mismo MAPSET, el nombre del mapa debe corresponder a los primeros caracteres del MAPSET y un consecutivo
* Nombre de las variables del mapa: M*XXXXXX, XXXXXX*: Nombre de la variable
* Todos los campos para ingreso de datos tienen definido en la posición siguiente un campo ASKIP de longitud 1 para que el sistema salte al siguiente campo una vez completado
* COPY de mapa simbólico – Ubicación: P390.CICS.COPY
* Diseño del mapa:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Posición | Tamaño | Color | Formato | Observaciones |
| Transacción | 1,1 | 4 | Verde | XX99 | Ej SN01 |
| Programa | 1,7 | 9 | Verde |  | Nombre del programa |
| MFecha | 1,68 | 10 | Verde | AAA/MM/DD | Campo alimentado con el nombre del programa |
| Titulo | 3,  centrado |  | Verde |  |  |
| Línea de separación | 4,1 | 79 | Verde | ------ | Se omite si se requiere el espacio |
| Descripción campos |  |  | Verde |  |  |
| Cuerpo del mapa | | | | | |
| Campos desplegados | 6…20, n |  | Amarillo |  |  |
| Campos de ingreso de datos | 6…20, n |  | Turquesa |  |  |
| Fin cuerpo del mapa | | | | | |
| Mensajes de usuario | 21-22, 6 | 70 | Neutro Intermitente si requiere alerta |  | Se utiliza la línea 23 en caso de ser necesario. |
| Teclas de función | 23-24, 2 | 76 | Verde |  |  |

## PROGRAMAS

* Ubicación de programas: P390.CICS.SOURCE
* Nombre Programa: P*YYXXX*99
  + Longitud: 8 caracteres
  + *YY*-Identificador de la aplicación
  + *XXX-*Descripción de la función
  + 99 - Consecutivo de los programas
  + P*YY*MEN99 Para los programas que manejan menús
* Nombre de Rutinas: R*YYXXXXX*
  + YY-Identificador de la aplicación
  + XXXXX-Descripción de la tabla para el caso de que sea encapsulador, o descripción de la función realizada. En caso de que el encapsulador sea llamado por varias aplicaciones, no tendrá identificador de la aplicación
* COPYs de código – Ubicación: P390.CICS.COPY Uso de copy para código genérico, tal como COMMAREA de los programas pseudoconversacionales
* Nombres de variables:

WS- Variables de trabajo Working-storage section

WC- Constantes

WC-CUR- Posición del cursor

SW- Banderas o Switches

CA- Definición de Campos CA: Variable de COMMAREA

WP-COMMAREA: área de trabajo de la COMMAREA

LK- Definición de campos que redefinen la COMMAREA en un llamado a o desde un programa (son parte de la LINKAGE SECTION). Para los parámetros provenientes de encapsulados, debe contener el prefijo de los campos de la tabla LK-XX-

* Nombres de los párrafos
  + Máximo 30 caracteres

0000-PRINCIPAL.

Invoca a los demás componentes de la estructura del programa

1000-INICIO.

Inicio de programa

2000-PROCESO-TRANSACCION.

Manejo de ciclos del programa pseudoconversacional

2100 - 6990

Todos los párrafos relacionados con el objetivo del programa

7000-ENVIO-MAPA.

7100-RECIBO-MAPA.

7800-RETORNO-TRANSACCION.

7900-7990 - Paso de control a otro programa (LINK-XCTL)

9000-MANEJO-ERROR.

9999-FIN.

* Cada párrafo debe terminar con . punto en línea aparte para evitar errores
* Evitar el uso de PERFORM THRU
* Toda la funcionalidad de acceso a la base de datos será realizada mediante rutinas encapsuladoras, con las siguientes características:
* Nombre: RXXXXXNN
* Cada tabla tendrá una rutina para suplir todas las funciones requeridas
* Si se requiere unir varias tablas, “join”, esta unión se hará desde el programa,

el join lo debe hacer el programa de la lógica del negocio, involucrando los llamados a las tablas que necesite mediante cada uno de los encapsuladores relacionados

## ELEMENTOS DB2

SPUFI

* SPU999 - Nombre de los archivos SPUFI
* Lo SPUFI deben estar localizados en P390.DB2.SPUFI

DCLGEN

* Ubicación: P390.DB2.DCLGEN Tiene los copies de las tablas de la base de datos, deben tener prefijo para los campos XX- que identifiquen la tabla y la estructura XX-NombreTabla
* Debe crearse un Table Space para cada tabla e identificarse con los primeros 6 caracteres del nombre de la tabla y el sufijo TS
* Los nombres de los campos de las tablas deben ser lo más significativo posible

BIND

* Ubicación: P390.DB2.BIND
* Bind Package BPK*YYXXX* – *YY* Sistema *XXX* prefijo del programa o tabla de la BD, ej BPKSNUSU Bind del Package para el Sistema de Notas del encapsulador de la tabla de usuarios
* Bind Plan BPL*YYYYY* – ej: BPLNOTAS Bind del Plan de Sistema de Notas
* Nombre Package PKYYYYY - PKNOTAS - sufijo del programa
* Nombre Plan Pl*YYYYY*  - PLNOTAS Iniciales del nombre del subsistema

## AMBIENTE CICS

* + Grupo: G*XXXXX* – *XXXXX* Sistema ej. GNOTAS
  + Transacción: YY99 YY Sistema, ej SN01
  + DB2ENTRY: E*YYYYY* – *YYYYY* sistema, ej. ENOTAS
  + DB2TRANS: T*YYYYYY* – *YYYYYY* Nombre del programa

## CONTROL DE VERSIONES

|  |  |  |
| --- | --- | --- |
| Fecha | Descripción | Observaciones |
| 10 septiembre 2024 | Mejora en formato de documento e incorpora estándar de numeración y nombres de párrafos |  |
|  |  |  |