**Documento de Referencia para Encapsulador COBOL PCICS56**

**1.Descripción General**

**1.1 Nombre del Encapsulador:** Encapsulador de Operaciones CRUD para Tabla Estudiantes  
**1.2 Propósito:** Este encapsulador permite realizar operaciones de creación de Cursores, inserción, selección, actualización y eliminación en la tabla Estudiantes de la base de datos. Las operaciones se determinan según los valores de TIPO y SUB-TIPO recibidos en la DFHCOMMAREA.

**2. Parámetros de Entrada**

* 1. **TIPO**: Define la operación a realizar.
* INSERT: Inserta un nuevo registro.
* SELECT: Selecciona uno o varios registros.
* UPDATE: Actualiza un registro existente.
* DELETE: Elimina un registro existente.
* CURSOR: Crea un cursor cuando el criterio de búsqueda trae más de un registro
  1. **SUB-TIPO**: Define una secuencia específica para la operación. Por ejemplo:
* 01: Operación básica para el tipo seleccionado.

**3.Campos Utilizados en la Operación**

* **TIPOUSUA**: Código del tipo de usuario para filtrar los registros.
* **CED**:Mumero de Cedula del estudiante
* **CLAVE**: Clave del estudiante.
* **NOM**: Nombre del estudiante.
* **APE**: Apellido del estudiante.
* **DIR**: Dirección del estudiante.
* **CARGO**:Cargo del Estudiante
* **ESTADO**:Estado del Estudiante

**4. Descripción del Código**

El encapsulador utiliza secciones específicas para cada tipo de operación. A continuación, se muestra cada sección:

\*

**4.1 INSERT.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* INSERTA LOS DATOS EN LA TABLA \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**4.1.1 0110-INSERT-01.**

MOVE TIPOUSUA TO US-TIPOUSUA

MOVE CED TO US-DNI

MOVE CLAVE TO US-CLAVE

MOVE NOM TO US-NOMBRE

MOVE APE TO US-APELLIDOS

MOVE DIR TO US-DIRECCION

MOVE CARGO TO US-CARGO

MOVE ESTADO TO US-ESTADO

EXEC SQL

INSERT INTO

USUARIOS(TIPOUSUA,DNI,CLAVE,NOMBRE,APELLIDOS,

DIRECCION,CARGO,ESTADO)

VALUES(:US-TIPOUSUA,:US-DNI,:US-CLAVE,:US-NOMBRE,

:US-APELLIDOS,:US-DIRECCION,:US-CARGO,:US-ESTADO)

END-EXEC

IF SQLCODE = 0

MOVE '0' TO EXISTE

ELSE

MOVE '9' TO EXISTE

END-IF

**4.1.2 Ejemplo de Uso**

Se mueven los siguientes campos al MI-COMMAREA MOVE para mas adelante en el

Link enviar la información que necesita el Encapsulador

MOVE 'ESTUD' TO TIPOUSUA TIPOUSUA

MOVE CEDI TO CED Cedula

MOVE APEI(1:3) TO CLAVE(1:3) Clave

MOVE NOMI(1:3) TO CLAVE(4:3)

MOVE ' ' TO CLAVE(7:2)

MOVE NOMI TO NOM Nombre

MOVE APEI TO APE Apellido

MOVE DIRI TO DIR Direccion

MOVE 'ESTUDIANTE' TO CARGO Cargo

MOVE 'A' TO ESTADO Estado

MOVE SPACES TO EXISTE Estado de la transaccion

MOVE 'INSERT' TO TIPO Tipo de operación a ejecutar en encapsulador

MOVE '01' TO SUB-TIPO Secuencia de la operacion

**PERFORM 9000-PROCESE-CRUD**

IF EXISTE = '0' THEN

INSERT exitoso

ELSE

INSERT con problemas

END-IF .

**4.1.3 Manejo de Errores**

Error al insertar un registro duplicado: Verifique que el valor de DNI no exista ya en la base de datos.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* REALIZA EL UPDATE \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**4.2 UPDATE.**

**4.2.1**  **0120-UPDATE-01.**

MOVE TIPOUSUA TO US-TIPOUSUA

MOVE CED TO US-DNI

MOVE NOM TO US-NOMBRE

MOVE APE TO US-APELLIDOS

MOVE DIR TO US-DIRECCION

EXEC SQL

UPDATE USUARIOS

SET CLAVE = :US-CLAVE ,

NOMBRE = :US-NOMBRE,

APELLIDOS = :US-APELLIDOS,

DIRECCION = :US-DIRECCION

WHERE

TIPOUSUA = :US-TIPOUSUA AND

DNI = :US-DNI

END-EXEC

IF SQLCODE = 0

MOVE '0' TO EXISTE

ELSE

MOVE '9' TO EXISTE

END-IF

**4.2.2 Ejemplo de Uso**

Se mueven los siguientes campos al MI-COMMAREA MOVE para mas adelante en el

Link enviar la información que necesita el Encapsulador

MOVE 'ESTUD' TO TIPOUSUA TIPOUSUA

MOVE APEI(1:3) TO CLAVE(1:3) Clave

MOVE NOMI(1:3) TO CLAVE(4:3)

MOVE ' ' TO CLAVE(7:2)

MOVE CEDI TO CED Cedula

MOVE NOMI TO NOM Nombre

MOVE APEI TO APE Apellido

MOVE DIRI TO DIR Direccion

MOVE SPACES TO EXISTE Estado de la transaccion

MOVE ‘UPDATE' TO TIPO Tipo de operación a ejecutar en encapsulador

MOVE ‘01' TO SUB-TIPO Secuencia de la operacion

**PERFORM 9000-PROCESE-CRUD**

IF EXISTE = '0' THEN

UPDATE exitoso

ELSE

Update con problemas

END-IF .

**4.2.3 Manejo de Errores**

* + Error al actualizar un registro que no existe: Verifique que el valor de DNI ya exista ya en la base de datos.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* LEE UN SOLO REGISTRO CON NUMERO CLAVE \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

**4.3 SELECT**

**4.3 .1 0130-SELECT-01.**

MOVE TIPOUSUA TO US-TIPOUSUA

MOVE CED TO US-DNI

EXEC SQL

SELECT

DNI,

NOMBRE,

APELLIDOS,

DIRECCION

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :US-TIPOUSUA AND

DNI = :US-DNI

END-EXEC

IF SQLCODE = +100

MOVE '1' TO EXISTE

MOVE SPACES TO NOM

MOVE SPACES TO APE

MOVE SPACES TO DIR

ELSE

IF SQLCODE = 0

MOVE '0' TO EXISTE

MOVE US-DNI TO CED

MOVE US-NOMBRE TO NOM

MOVE US-APELLIDOS TO APE

MOVE US-DIRECCION TO DIR

ELSE

MOVE '9' TO EXISTE

MOVE SPACES TO CED

MOVE SPACES TO NOM

MOVE SPACES TO APE

MOVE SPACES TO DIR

END-IF

END-IF

**4.3.2 Ejemplo de Uso**

Se mueven los siguientes campos al MI-COMMAREA MOVE para mas adelante en el Link enviar la información que necesita el Encapsulador

MOVE 'ESTUD' TO TIPOUSUA TIPOUSUA

MOVE CEDI TO CED Cedula

MOVE 'SELECT' TO TIPO Tipo de operación a ejecutar en encapsulador

MOVE '01' TO SUB-TIPO Secuencia de la operacion

**PERFORM 9000-PROCESE-CRUD**

IF EXISTE = '0' THEN

SELECT exitoso

ELSE

SELECT con problemas

END-IF .

**4.3.3 Manejo de Errores**

* + Error al leer un registro que no existe: Verifique que el valor de DNI ya exista ya en la base de datos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* BORRA UN REGISTRO CON EL NUMERO DE LA CLAVE \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

**4.4 Delete**

**4.4.1 0140-DELETE-01.**

MOVE TIPOUSUA TO US-TIPOUSUA

MOVE CED TO US-DNI

EXEC SQL

DELETE FROM USUARIOS

WHERE

TIPOUSUA = :US-TIPOUSUA AND

DNI = :US-DNI

END-EXEC

IF SQLCODE = 0

MOVE '0' TO EXISTE

ELSE

MOVE '9' TO EXISTE

END-IF

**4.4.2 Ejemplo de Uso**

Se mueven los siguientes campos al MI-COMMAREA MOVE para mas adelante en el Link enviar la información que necesita el Encapsulador

MOVE 'ESTUD' TO TIPOUSUA TIPOUSUA

MOVE CEDI TO CED Cedula

MOVE 'DELETE' TO TIPO Tipo de operación a ejecutar en encapsulador

MOVE '01' TO SUB-TIPO Secuencia de la operacion

**PERFORM 9000-PROCESE-CRUD**

IF EXISTE = '0' THEN

SELECT exitoso

ELSE

SELECT con problemas

END-IF

.

**4.4.3 Manejo de Errores**

* + Error al leer un registro que no existe: Verifique que el valor de DNI ya exista ya en la base de datos.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Manejo de Cursores \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**4.5 Cursores.**

**4.5.1 Cursor: CSR1.**

**4.5.1.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que

cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los

registros se ordenan por DNI en orden ascendente.

EXEC SQL

DECLARE CSR1 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY DNI ASC

END-EXEC

**4.5.1.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.1.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.1.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal Con nombre

WS-NOMBRECOLA ES KCICS03

EXEC SQL

FETCH CSR1

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

.

**4.5.1.5** **Manejo de cola Temporal:**

EXEC SQL

FETCH CSR1

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.1.6 Código del Cursor: CSR1**

* **Tipo:** CURSOR
* **Sub-Tipo:** 01

**4.5.1.7 Ejemplo de Uso**

* **Tipo:** CURSOR
* **Sub-Tipo:** 01

**4.5.1.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.2 Cursor: CSR2**

**4.5.2.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los registros se ordenan por DNI en orden descendente.

EXEC SQL

DECLARE CSR2 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY DNI DESC

END-EXEC

**4.5.2.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.2.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.2.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.2.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR2

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**Uso:**

**4.5.2.6 Código del Cursor: CSR2**

* **Tipo:** CURSOR
* **Sub-Tipo:** 02

**4.5.2.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '02' TO SUB-TIPO

PERFORM 9000-PROCESE-CRUD

**4.5.2.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.3 Cursor: CS3**

**4.5.3.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los registros se ordenan por NOMBRE en orden ascendente.

EXEC SQL

DECLARE CSR3 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY NOMBRE ASC

END-EXEC

**4.5.3.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.3.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.3.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal Con nombre

WS-NOMBRECOLA ES KCICS03

**4.5.3.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR3

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.3.6 Código del Cursor: CSR3**

* **Tipo:** CURSOR
* **Sub-Tipo:** 03

**4.5.3.7 Ejemplos de uso:**

MOVE 'CURSOR' TO TIPO

MOVE '03' TO SUB-TIPO

PERFORM 9000-PROCESE-CRUD

**4.5.3.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.4 Cursor: CSR4**

**4.5.4.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los registros se ordenan por NOMBRE en orden descendente.

EXEC SQL

DECLARE CSR4 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY NOMBRE DESC

END-EXEC

**4.5.4.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.4.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.4.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.4.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR4

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.4.6 Código del Cursor: CSR4**

* **Tipo:** CURSOR
* **Sub-Tipo:** 04

**4.5.4.7 Ejemplo de Uso:**

MOVE 'CURSOR' TO TIPO

MOVE '04' TO SUB-TIPO

PERFORM 9000-PROCESE-CRUD

**4.5.4.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.5** **Cursor: CS5**

**4.5.5.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los registros se ordenan por DIRECCION en orden ascendente.

EXEC SQL

DECLARE CSR5 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY NOMBRE ASC

END-EXEC

**4.5.5.2 Campos de Entrada.**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.5.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.5.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal Con nombre

WS-NOMBRECOLA ES KCICS03

**4.5.5.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR5

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.5.6 Código del Cursor: CSR5**

* **Tipo:** CURSOR
* **Sub-Tipo:** 05

**4.5.5.7 Ejemplos de uso:**

MOVE 'CURSOR' TO TIPO

MOVE '05' TO SUB-TIPO

PERFORM 9000-PROCESE-CRUD

**4.5.5.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.6 Cursor: CSR6**

**4.5.6.1 Descripción:** Este cursor recupera todos los registros de la tabla USUARIOS que cumplan con la condición de tener un valor específico en el campo TIPOUSUA. Los registros se ordenan por DIRECCION en orden descendente.

EXEC SQL

DECLARE CSR6 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

TIPOUSUA = :TIPOUSUA

ORDER BY NOMBRE DESC

END-EXEC

**4.5.6.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.

**4.5.6.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.6.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.6.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR6

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

.

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**Uso:**

**4.5.6.6.Código del Cursor: CSR6**

* **Tipo:** CURSOR
* **Sub-Tipo:** 06

**4.5.6.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '06' TO SUB-TIPO

PERFORM 9000-PROCESE-CRUD

**4.5.6.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.7 Cursor:CSR7**

**4.5.7.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable :MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por DNI en orden Ascendente

EXEC SQL

DECLARE CSR7 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY DNI ASC

END-EXEC

EXEC SQL

**4.5.7.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.7.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.7.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.7.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR7

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

.**4.5.7.6**  **Código del Cursor: CSR7**

* **Tipo:** CURSOR
* **Sub-Tipo:** 07
* **Mlike:**Criterio de busqueda

**4.5.7.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '07' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.7.8 Consideraciones:**

* Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.
* El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.8 Cursor: CSR8**

**4.5.8.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable :MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por DNI en orden Descente

EXEC SQL

DECLARE CSR8 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY DNI DESC

END-EXEC

EXEC SQL

**4.5.8.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.8.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.8.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.8.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR8

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.8.6 Código del Cursor: CSR8**

* **Tipo:** CURSOR
* **Sub-Tipo:** 08
* **Mlike:**Criterio de busqueda

**4.5.8.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '08' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.8.8 Consideraciones:**

Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.

El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.9 Cursor:CSR9**

**4.5.9.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable: MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por NOMBRE en orden Ascendente

EXEC SQL

DECLARE CSR9 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY NOMBRE ASC

END-EXEC

EXEC SQL

**4.5.9.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.9.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.9.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.9.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR9

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.9.6 Código del Cursor: CSR9**

* **Tipo:** CURSOR
* **Sub-Tipo:** 09
* **Mlike:**Criterio de busqueda

**4.5.9.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '09' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.9.8 Consideraciones:**

Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.

El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos

**4.5.10 Cursor:CSR10**

**4.5.10.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable: MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por NOMBRE en orden Descendente

EXEC SQL

DECLARE CSR10 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY NOMBRE DESC

END-EXEC

EXEC SQL

**4.5.10.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.10.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.10.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.10.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR10

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.10.6**  **Código del Cursor: CSR10**

* **Tipo:** CURSOR
* **Sub-Tipo:** 10
* **Mlike:**Criterio de busqueda

**4.5.10.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '10' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.10.8 Consideraciones:**

Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.

El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.11 Cursor:CSR11**

**4.5.11.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable: MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por APELLIDO en orden Ascendente

EXEC SQL

DECLARE CSR11 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY APELLIDO ASC

END-EXEC

EXEC SQL

**4.5.11.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.11.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.11.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.11.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR11

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.11.6 Código del Cursor: CSR11**

* **Tipo:** CURSOR
* **Sub-Tipo:** 11
* **Mlike:**Criterio de busqueda

**4.5.11.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '11' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.11.8 Consideraciones:**

Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.

El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.5.12 Cursor:CSR12**

**4.5.12.1 Descripción:** Este cursor está buscando coincidencias parciales (LIKE) en varios campos (DNI, NOMBRE, APELLIDOS, DIRECCION) para el valor proporcionado en la variable: MLIKE. Además, está filtrando los resultados por un tipo de usuario específico (TIPOUSUA), los registros se ordenan por APELLIDO en orden Descendente

EXEC SQL

DECLARE CSR12 CURSOR FOR

SELECT

DNI

,NOMBRE

,APELLIDOS

,DIRECCION

FROM

USUARIOS

WHERE

(DNI LIKE :MLIKE

OR UPPER(NOMBRE) LIKE :MLIKE

OR UPPER(APELLIDOS) LIKE :MLIKE

OR UPPER(DIRECCION) LIKE :MLIKE) AND

(TIPOUSUA = :TIPOUSUA)

ORDER BY APELLIDO DESC

END-EXEC

EXEC SQL

**4.5.12.2 Campos de Entrada:**

* TIPOUSUA: Código del tipo de usuario para filtrar los registros.
* MLIKE: Coincidencias a buscar

**4.5.12.3 Campos de Salida:**

* DNI: Número de identificación del usuario.
* NOMBRE: Nombre del usuario.
* APELLIDOS: Apellidos del usuario.
* DIRECCION: Dirección del usuario.

**4.5.12.4 Operación de FETCH:**

* Por cada registro recuperado con el FETCH, los valores se mueven a variables intermedias y luego se escriben en una cola temporal . Nombre asignado a WS-NOMBRECOLA ES KCICS03

**4.5.12.5 Manejo de cola Temporal:**

EXEC SQL

FETCH CSR12

INTO

:US-DNI

,:US-NOMBRE

,:US-APELLIDOS

,:US-DIRECCION

END-EXEC

IF SQLCODE = +100

MOVE 'Y' TO WS-EOF

END-IF

IF SQLCODE = 0

ADD 1 TO WS-CONTADOR-REGISTRO

MOVE US-DNI TO C-DNI

MOVE US-NOMBRE TO C-NOMBRE

MOVE US-APELLIDOS TO C-APELLIDOS

MOVE US-DIRECCION TO C-DIRECCION

EXEC CICS WRITEQ TS

QUEUE (WS-NOMBRECOLA)

FROM (COLA-USUARIOS)

END-EXEC

**4.5.12.6 Código del Cursor: CSR10**

* **Tipo:** CURSOR
* **Sub-Tipo:** 12
* **Mlike:**Criterio de busqueda

**4.5.12.7 Ejemplos de uso**

MOVE 'CURSOR' TO TIPO

MOVE '12' TO SUB-TIPO

Move Criterio de búsqueda to MLIKE

PERFORM 9000-PROCESE-CRUD

**4.5.12.8 Consideraciones:**

Asegúrate de que el campo TIPOUSUA y el campo MLIKE esté correctamente poblado antes de ejecutar el cursor.

El encapsulador se encarga de realizar el open , el fetch y el close del cursor y liberarlos recursos.

**4.6 9000-PROCESE-CRUD**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* LLAMAMOS AL PROGRAMA QUE REALIZA LAS OPERACIONES DEL CRUD\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MOVE MI-COMMAREA TO RC-COMMAREA

MOVE LOW-VALUES TO MI-COMMAREA

MOVE 'LINK' TO TRANS

EXEC CICS LINK

PROGRAM('PCICS56')

COMMAREA(RC-COMMAREA)

LENGTH(LENGTH OF MI-COMMAREA)

RESP(RESPUESTA)

END-EXEC

IF RESPUESTA = DFHRESP(NORMAL) THEN

MOVE RC-COMMAREA TO MI-COMMAREA

END-IF

.