

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

OPERACIONES ARITMETICAS	6
ADD (Sentencia Sumar)	6
SUB (Sentencia Restar)	7
MULT (Sentencia Multiplicar)	8
DIV (Sentencia Dividir)	9
MVR (Sentencia para recuperar el Resto de la división)	10
MOVE (Sentencia para asignar el valor de una variable a otra)	11
EVAL (Función que resuelve una expresión aritmética)	12
AJTRED (Función que aplica redondeo)	13
APLPCJ (Función aplicar tanto por ciento)	14
APLPRO (Función para aplicar un prorrateo a un importe).	15
INTLIN (Función para calcular Interes $((C * R * T) / (100 * UT))$)	16
INTEXP (Función para calcular Interes $(C * (1 + (R / 100)^N) - C)$)	17
CLCPOT (Función para calcular una potencia)	18
DIGVER (Función para calcular dígito verificador)	19
CVTASA (Función para convertir tasas)	20
GENCUI (Función para generar un CUIT)	21
OPERACIONES CONDICIONADAS	22
IF (Sentencia para condicionar ejecución)	22
IFR (Sentencia para condicionar ejecución por rango)	23
DOUNTIL (Sentencia para condicionar ejecución <i>hasta que</i>)	24
DOWHILE (Sentencia para condicionar ejecución <i>mientras que</i>)	25
DO (Sentencia para condicionar ejecución con índice)	26
WHEN (Sentencia para condicionar ejecución de grupos de instrucciones)	27
CASE (Sentencia para condicionar ejecución de grupos de rutinas de usuarios)	28
OPERACIONES DE BIFURCACIONES	29
GOTO (Sentencia para bifurcar a una etiqueta)	29
TAG (Sentencia para indicar una etiqueta)	30
LOOP (Sentencia para controlar otra iteración)	31
EXIT (Sentencia para salir de la iteración)	32
STREVN (Función para reiniciar la ejecución del evento)	33
ENDEVN (Función para finalizar la ejecución del evento)	34
FUNCIONES FECHA	35
WRKFEC (Función para trabajar con fechas)	35
CLCFEC (Función para calcular fechas)	36
CLCJUL (Función para calcular días julianos o gregorianos desde fecha)	37

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FECALF (Función para convertir una fecha en alfabética)	38
CLCDIA (Función para calcular días entre dos fechas)	39
VALFEC (Función para validar una fecha)	40
VALHAB (Función para validar una fecha habil)	41
EXTIME (Función para recuperar fecha y hora actual)	42
FUNCIONES DE LECTURA	43
LEEREG (Función para acceder a un registro de un archivo)	43
LEECPT (Función para acceder a un registro de comprobante)	44
LEESDO (Función para acceder a un registro de Saldo)	45
LEEGRP (Función para acceder a un grupo de registros de un archivo)	46
LEETXT (Función para leer un registro de información variable)	47
SQLLEE (Función para leer un registro por SQL)	48
FUNCIONES DE CONTROL	49
CTLOPE (Función para controlar operatividad de una cuenta)	49
CTLCNF (Función para controlar cheque conformado)	50
CTLONP (Función para controlar Orden de no Pagar)	51
CTLAPL (Función para controlar aplicabilidad de un importe)	52
CTLIMP (Función para controlar un importe)	53
CTLEST (Función para controlar el estado de un comprobante)	54
CTLACC (Función para controlar la acción de un comprobante)	55
CTLSNC (Función para controlar sincronización del mensaje)	56
FUNCIONES DE IMPRESION	57
OPNPRT (Función para habilitar un archivo de impresión)	57
PRTLIN (Función para imprimir una línea con información)	58
CLOPRT (Función para cerrar un archivo de impresión)	59
PRTFOR (Función para imprimir un formulario)	60
REPORT (Función para imprimir un reporte)	61
FUNCIONES DE PANTALLA	62
MNUEVN (Función para visualizar un menú con una lista de eventos)	62
MNUKEY (Función para visualizar un menú de teclas rápidas)	63
MNUOPC (Función para visualizar un menú de opciones)	64
EXEFMT (Función para visualizar e ingresar datos por pantalla)	65
WRTFMT (Función para visualizar datos por pantalla)	66
INZSFL (Inicializar un Sub File de consulta)	67
LOADSF (Función para cargar datos en el Sub File de consulta)	68
FUNCIONES DE VARIABLES	69

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CONCAT (Función para concatenar dos variables en una tercera).	69
SUBSTR (Función para obtener una substring de un variable).	70
CVTDEC (Función para obtener campo número desde variable alfabética)	71
EDTDEC (Función para editar un campo numérico en una variable alfabética)	72
ASGVAL (Función para asignar valor carácter a una variable).	73
<i>FUNCIONES DE PROCESO</i>	74
RQRCTL (Función para controlar la ejecución de un paso de proceso previo).	74
RQRSTR (Función para iniciar un paso de proceso)	75
RQREND (Función para finalizar un paso de proceso)	76
RQRAUT (Función para solicitar autorización)	77
EXSR (Sentencia para ejecutar rutina definida por el usuario).	78
BEGSR (Sentencia para definir rutina de usuario)	79
EXFUN (Sentencia para ejecutar función definida por el usuario).	80
BEGFU (Sentencia para definir función de usuario).	81
CALL (Función para invocar a otro evento)	82
RETURN (Función para finalizar la ejecución de sentencias)	83
EXEC (Función para invocar a un comando del sistema operativo)	84
SNDMSG (Función para enviar mensajes al cliente)	85
WRTMSG (Función para visualizar mensajes y continuar)	86
EXEMSG (Función para visualizar mensajes y esperar)	87
STRDBG(Función para iniciar depuracion)	88
<i>FUNCIONES DE ARCHIVO TIPO TEXTO</i>	89
OPNTXT (Función para abrir un archivo de información variable)	89
CLOTXT (Función para cerrar un archivo de información variable)	90
DEFREG (Función para definir registro de información variable)	91
DEFTXT (Función para redefinir posiciones de los campos predefinidos)	92
MOVDAT (Función para mover datos al registro de información variable)	93
<i>FUNCIONES INCORPORADAS</i>	94
NMWORK (Función para generar un nombre de trabajo aleatorio)	94
IMPLET (Función para convertir un importe en letras)	95
MOVCHQ (Función para mover un ítem del vector de cheques)	96
RTVTAS (Función para recuperar una tasa)	97
PLFVTO (Función para validar fecha de vencimiento y plazo de Plazo Fijo)	98
<i>FUNCIONES DE GRABACIÓN / ACTUALIZACION</i>	99
UPDREG (Función para actualizar el registro de un archivo)	99
WRTREG (Función para grabar el registro de un archivo)	100

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

WRTTXT (Función para grabar el registro de un archivo de información variable) _____	101
GRAMOV (Función que graba un movimiento) _____	102
GRACPT (Función que graba un comprobante) _____	103
GRAPLF (Función que graba un certificado de plazo fijo) _____	104
APLIMP (Función para aplicar un importe al saldo) _____	105
COMMIT (Función para definir una unidad de procesamiento) _____	106
ROLLBACK (Función para anular operaciones) _____	107
NOMECLATURA y DEFINICIONES _____	108
Comentario dentro de sentencias (/ * comentario */) _____	108
Literal Numérica (0/9, -, +) _____	108
Literal Alfanumérica (“caracteres”) _____	108
Campos de tablas y Variables (&campo, %campo) _____	108
Palabras Reservadas (*AAAAAAAAAAA) _____	108
Variables de uso general _____	108
Etiqueta (A/Z) _____	108
FORMATOS DE PANTALLA _____	109
Formato = *CTAREF (Cuenta referenciada) _____	109
Formato = *CTATRN (Cuenta operativa de la transacción) _____	110
Formato = *FEC1 (Fecha 1 cuadrante A) _____	111
Formato = *FEC1 (Fecha 1 cuadrante B) _____	112
Formato = *FEC1 (Fecha 1 cuadrante C) _____	113
Formato = *FEC2 (Fecha 2 cuadrante A) _____	114
Formato = *FEC2 (Fecha 2 cuadrante B) _____	115
Formato = *FEC2 (Fecha 2 cuadrante C) _____	116
Formato = *FEC3 (Fecha 3 cuadrante A) _____	117
Formato = *FEC3 (Fecha 3 cuadrante B) _____	118
Formato = *FEC3 (Fecha 3 cuadrante C) _____	119
Formato = *CPT1 (Comprobante 1 cuadrante A) _____	120
Formato = *CPT1 (Comprobante 1 cuadrante B) _____	121
Formato = *CPT1 (Comprobante 1 cuadrante C) _____	122
Formato = *CPT2 (Comprobante 2 cuadrante A) _____	123
Formato = *CPT2 (Comprobante 2 cuadrante B) _____	124
Formato = *CPT2 (Comprobante 2 cuadrante C) _____	125
Formato = *CPT3 (Comprobante 3 cuadrante A) _____	126
Formato = *CPT3 (Comprobante 3 cuadrante B) _____	127
Formato = *CPT3 (Comprobante 3 cuadrante C) _____	128
Formato = *TAS1 (Tasa 1 cuadrante A) _____	129

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS1 (Tasa 1 cuadrante B)	130
Formato = *TAS1 (Tasa 1 cuadrante C)	131
Formato = *TAS2 (Tasa 2 cuadrante A)	132
Formato = *TAS2 (Tasa 2 cuadrante B)	133
Formato = *TAS2 (Tasa 2 cuadrante C)	134
Formato = *TAS3 (Tasa 3 cuadrante A)	135
Formato = *TAS3 (Tasa 3 cuadrante B)	136
Formato = *TAS3 (Tasa 3 cuadrante C)	137
Formato = *IMP1 (Importe 1 cuadrante A)	138
Formato = *IMP1 (Importe 1 cuadrante B)	139
Formato = *IMP1 (Importe 1 cuadrante C)	140
Formato = *IMP2 (Importe 2 cuadrante A)	141
Formato = *IMP2 (Importe 2 cuadrante B)	142
Formato = *IMP2 (Importe 2 cuadrante C)	143
Formato = *IMP3 (Importe 3 cuadrante A)	144
Formato = *IMP3 (Importe 3 cuadrante B)	145
Formato = *IMP3 (Importe 3 cuadrante C)	146
Formato = *CAN (Cantidad cuadrante A)	147
Formato = *CAN (Cantidad cuadrante B)	148
Formato = *CAN (Cantidad cuadrante C)	149
Formato = *SDO (SalDOS)	150
Formato = *PLF1 (tres primeros integrantes de la cuenta)	151
Formato = *PLF2 (Determina fecha de vencimiento)	152
Formato = *PLF3 (tasa, representante con documento)	153
Formato = *PLF4 (Interés, Retenciones y neto)	154
Formato = *IDCLI (Identificación del cliente)	155
Formato = Invocado por la función PRTFOR para impresión de comprob. con numeración.	156

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

OPERACIONES ARITMETICAS

ADD (Sentencia Sumar)

La sentencia ADD causa que dos operadores numéricos puedan ser sumados y el resultado almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

$$\text{ADD} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Identificador-2} \\ \text{Identificador-3} \end{array} \right\}$$

- *El identificador-1 o el literal-1 es adicionado al Identificador-2 o al literal-2 y el resultado es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

SUB (Sentencia Restar)

La sentencia SUB causa que dos operadores numéricos puedan ser restados y el resultado almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

SUB { Identificador-1 } { Identificador-2 } { Identificador-1
Literal-1 } { Literal-2 } { Identificador-2
Identificador-3 }

- *El identificador-1 o el literal-1 es restado al Identificador-2 o al literal-2 y el resultado es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MULT (Sentencia Multiplicar)

La sentencia MULT causa que dos operadores numéricos puedan ser multiplicados y el resultado almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

$$\text{MULT} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Identificador-2} \\ \text{Identificador-3} \end{array} \right\}$$

- *El identificador-1 o el literal-1 multiplica al Identificador-2 o al literal-2 y el resultado es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DIV (Sentencia Dividir)

La sentencia DIV causa que un operador numérico sea dividido por otro y el resultado almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

DIV { **Identificador-1** } { **Identificador-2** } { **Identificador-1**
Literal-1 } { **Literal-2** } { **Identificador-2**
Identificador-3 }

- *El identificador-1 o el literal-1 es dividido por el Identificador-2 o al literal-2 y el resultado es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MVR (Sentencia para recuperar el Resto de la división)

La sentencia MVR causa que un operador numérico sea dividido por otro y el resto almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

$$\text{MVR} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Identificador-2} \\ \text{Identificador-3} \end{array} \right\}$$

- *El identificador-1 o el literal-1 es dividido por el Identificador-2 o al literal-2 y el resto es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MOVE (Sentencia para asignar el valor de una variable a otra)

La sentencia MOVE transfiere los datos contenidos en variables o constantes, a una variable distinta o igual a la utilizada dentro de la sentencia.

Formato

MOVE $\left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Identificador-2} \end{array} \right\}$

➤ *El contenido del identificador-1 o el literal-1 es transferido al Identificador-1 o al Identificador-2.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EVAL (Función que resuelve una expresión aritmética)

La función EVAL calcula el valor de una expresión aritmética y asigna el resultado en una variable numérica.

Formato

$$\text{EVAL} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} = \left\{ \begin{array}{l} \text{Expresión-aritmética-1} \\ \text{Identificador-2} \\ \text{Literal-1} \end{array} \right\}$$

- *Al identificador-1 le asigna el resultado de la resolución de la expresión.aritmética-1 o el contenido del Identificador-2 o el valor del Literal-1.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

AJTRED (Función que aplica redondeo)

La función AJTRED aproxima el dígito involucrado a un valor superior en caso de ser 5 o mayor, y aproxima a un valor menor en caso de ser menor de 5.

Formato

AJTRED { Identificador-1 } { Literal-1 } { *U } { *5 } { *2 } { *T } { Literal-2 } { Identificador-1 } { Identificador-2 }

- Al identificador-1 se le aplica el redondeo según el número de dígito definido en el Literal-2 y el resultado es almacenado en Identificador-1 o Identificador-2.
- Forma de redondear:
 - *U – redondea a la unidad siguiente
 - *5 – redondea utilizando múltiplo de 5
 - *2 – redondea utilizando múltiplo de 2
 - *T – produce truncamiento

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

APLPRO (Función para aplicar un prorrateo a un importe).

La función APLPRO toma un valor base, la sumatoria de los valores bases y utilizando el valor a distribuir, obtiene el valor prorrateado.

Formato

APLPRO { Identificador-1 } { Identificador-2 } { Identificador-3 } { Identificador-4 }
 { Literal-1 } { Literal-2 } { Literal-3 } { }

- *El Identificador-1 o Literal-1 debe contener el valor a prorratear.*
- *El Identificador-2 o Literal-2 debe contener el valor base al cual se le aplicará el prorrateo.*
- *El Identificador-3 o Literal-3 debe contener la sumatoria de los valores base.*
- *El Identificador-4 almacenará el valor prorrateado.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

INTLIN (Función para calcular Interes ((C *R * T) / (100 * UT)

La función INTLIN calcula interés simple.

Formato

INTLIN { Identificador-1 } { Identificador-2 } { Identificador-3 }
 { Literal-1 } { Literal-2 } { Literal-3 }

 { Identificador-4 } { Identificador-5 }
 { Literal-4 } { }
 { *D365 *M }

- *El Identificador-1 o Literal-1 debe contener el Capital.*
- *El Identificador-2 o Literal-2 debe contener la Tasa.*
- *El Identificador-3 o el Literal-3 debe contener los días, es decir el Tiempo.*
- *El Identificador-4 o Literal-4(número) debe contener Unidad de Tiempo o se puede utilizar: *D365 corresponde al 365 días del año y *M a 12 meses del año (1200).*
- *El Identificador-5 almacenará el resultado de la operación, es decir el Interes.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

INTEXP (Función para calcular Interes $(C * (1 + (R / 100)^N) - C)$)

La función INTEXP calcula interés exponencial.

Formato

INTEXP { **Identificador-1** } { **Identificador-2** } { **Identificador-3** } { **Identificador-4** }
 { **Literal-1** } { **Literal-2** } { **Literal-3** } { }

- *El Identificador-1 o Literal-1 debe contener el Capital.*
- *El Identificador-2 o Literal-2 debe contener la Tasa.*
- *El Identificador-3 o Literal-3 debe contener el período.*
- *El Identificador-4 almacenará el resultado de la operación, es decir el Interes.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CLCPOT (Función para calcular una potencia)

La función CLCPOT causa que un operador numérico pueda ser elevado a una determinada potencia y el resultado almacenado en una variable numérica distinta o en alguna de las utilizadas dentro de la sentencia.

Formato

$$\text{CLCPOT} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Identificador-2} \\ \text{Identificador-3} \end{array} \right\}$$

- *El Identificador-1 o el Literal-1(numérico) debe contener el valor para ser tratado.*
- *El Identificador-2 o al Literal-2(numérico) debe contener el exponente.*
- *El resultado es almacenado en el identificador referenciado en el tercer grupo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DIGVER (Función para calcular dígito verificador)

La función DIGVER calcula el dígito verificador o lo controla.

Formato

DIGVER { Identificador-1 } { Literal-1 } { *10 } { *11 } { *9713 } { *CAL } { *GEN } { *CTL } { Identificador-2 }

- *El Identificador-1 o Literal-1 debe contener el número para ser calculado o el número con el dígito para ser controlado.*
- *Modulo: *10 *11 *9713*
- *Tipo de Cálculo:*
 - *CAL Calcula dígito verificador*
 - *GEN Genera el dígito*
 - *CTL Controla dígito*
- *El Identificador-2 almacenará el resultado de la operación. Para el caso de controlar dígito dicho identificador tendrá 0= Dígito correcto y 1= Dígito incorrecto.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CVTASA (Función para convertir tasas)

La función CVTASA convierte una tasa a otra.

Formato

CVTASA	$\left\{ \begin{array}{ll} *TNAV & *TNAA \\ *TEMV & *TEMA \\ *TEAV & *TEAA \\ *TEDV & *TEDA \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\}$	$\left\{ \begin{array}{ll} *TNAV & *TNAA \\ *TEMV & *TEMA \\ *TEAV & *TEAA \\ *TEDV & *TEDA \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-3} \\ \text{Literal-3} \\ *NONE \end{array} \right\}$	

- *Tipo de Tasa, origen donde el subfijo V corresponde a vencida y el subfijo A corresponde a adelantada*
 - *TNA tasa nominal anual*
 - *TEM tasa efectiva mensual*
 - *TEA tasa efectiva anual*
 - *TED tasa efectiva diaria*
- *El Identificador-1 o Literal-1 debe contener el valor de la tasa origen.*
- *Tipo de Tasa, destino donde el subfijo V corresponde a vencida y el subfijo A corresponde a adelantada*
 - *TNA tasa nominal anual*
 - *TEM tasa efectiva mensual*
 - *TEA tasa efectiva anual*
 - *TED tasa efectiva diaria*
- *El Identificador-2 o Literal-2 debe contener el valor de la tasa destino..*
- *El Identificador-3 o Literal-3 debe indicar el plazo, necesario para algunas conversiones; en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

GENCUI (Función para generar un CUIT)

La función GENCUI genera un CUIT a partir de un número de 8 dígitos.

Formato

GENCUI { Identificador-1 } { Identificador-2 } { Identificador-3 }

Literal-1 *F *E *M

- *El Identificador-1 o Literal-1 debe contener el número que se utilizará para determinar el CUIT*
- *El Identificador-2 debe contener el sexo o se indica específicamente.*
- *Tipo de sexo:*
 - *F femenino*
 - *M masculino*
 - *E Empresa*
- *El Identificador-3 almacenará el resultado o sea el CUIT.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

OPERACIONES CONDICIONADAS

IF (Sentencia para condicionar ejecución)

La sentencia IF causa que una condición específica sea evaluada. La acciones del programa dependen del valor verdadero o falso de la condición.

Formato 1

IF	$\left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\}$	$\left\{ \begin{array}{l} *EQ *NE *LK \\ *GT *LT *NL \\ *GE *LE \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$
.....			
ENDIF			

- *El identificador-1 o el literal-1 es evaluado contra el Identificador-2 o Literal-2. La finalización de la sentencia IF está dada por un ENDIF.*

Formato 2

IF	$\left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\}$	$\left\{ \begin{array}{l} *EQ *NE *LK \\ *GT *LT *NL \\ *GE *LE \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$
.....			
ELSE			
.....			
ENDIF			

- *El identificador-1 o el literal-1 es evaluado contra el Identificador-2 o Literal-2. En caso de que el resultado sea verdadero ejecutaría las instrucciones definidas a continuación; si es falso ejecutaría las instrucciones definidas luego del ELSE. En ambos casos la finalización de la sentencia IF está dada por un ENDIF.*

Formato 3

IF	$\left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\}$	$\left\{ \begin{array}{l} *EQ *NE *LK \\ *GT *LT *NL \\ *GE *LE \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$
AND	$\left\{ \begin{array}{l} \text{Identificador-1..3} \\ \text{Literal-1..3} \end{array} \right\}$	$\left\{ \begin{array}{l} *EQ *NE *LK \\ *GT *LT *NL \\ *GE *LE \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Identificador-2/4} \\ \text{Literal-2/4} \end{array} \right\}$
.....			
ELSE			
.....			
ENDIF			

- *El identificador-1 o el literal-1 es evaluado contra el Identificador-2 o Literal-2, luego evalúa el conector lógico AND o OR y así sucesivamente produciendo anidamientos en donde está permitido el IF de IF. En caso de que el resultado sea verdadero ejecutaría las instrucciones definidas a continuación; si es falso ejecutaría las instrucciones definidas luego del ELSE. En ambos casos la*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

finalización de la sentencia IF está dada por un ENDIF. Debe existir un ENDIF por cada IF abierto.

IFR (Sentencia para condicionar ejecución por rango)

La sentencia IFR causa que una condición específica sea evaluada dentro de un rango. Las acciones del programa dependen del valor verdadero o falso de la condición.

Formato 1

```
IFR { Identificador-1 } { Identificador-2 } { Identificador-3 }
    { Literal-1       } { Literal-2       } { Literal-3       }
.....
ENDIF
```

- *El identificador-1 o el literal-1 es evaluado dentro del rango indicado por Identificador-2 o Literal-2(desde) e Identificador-3 o Literal-3. La finalización de la sentencia IF está dada por un ENDIF.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DOUNTIL (Sentencia para condicionar ejecución *hasta que*)

La sentencia DOUNTIL condiciona una iteración hasta que se cumple la condición especificada. Ejecuta y compara.

Formato

DOUNTIL { Identificador-1 } { *EQ *NE *LK } { Identificador-2 }
 { Literal-1 } { *GT *LT *NL } { Literal-2 }
 *GE *LE

ENDDU

- La instrucción siguiente al DOUNTIL se ejecuta incondicionalmente la primera vez, luego el identificador-1 o el literal-1 es evaluado contra el Identificador-2 o Literal-2, si la condición es verdadera continúa con la iteración, en caso de ser falsa se finaliza la sentencia DOUNTIL(dada por un ENDDU).

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DOWHILE (Sentencia para condicionar ejecución *mientras que*)

La sentencia DOWHILE condiciona una iteración mientras que se cumple la condición especificada. Compara y ejecuta.

Formato

DOWHILE { Identificador-1 } { *EQ *NE *LK } { Identificador-2 }
 { Literal-1 } { *GT *LT *NL } { Literal-2 }
 *GE *LE

ENDDW

- El identificador-1 o el literal-1 es evaluado contra el Identificador-2 o Literal-2, si la condición es verdadera ejecuta la siguiente instrucción, en caso de ser falsa se finaliza la sentencia DOWHILE(dada por un ENDDW).

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DO (Sentencia para condicionar ejecución con índice)

La sentencia DO es una iteración que ejecuta lo que se encuentre dentro del lazo mientras que el índice no halla alcanzado en valor especificado. Compara y ejecuta.

Formato

DO { Identificador-1 } { Identificador-2 } { Identificador-3 }
 { Literal-1 } { Literal-2 }
.....
ENDDO { Identificador-4 }
 { Literal-4 }

- *El Identificador-3 es incrementado por el Identificador-4 o el Literal-4 hasta que alcance el valor indicado en el Identificador-2 o el Literal-2 partiendo del valor contenido en el Identificador-1 o Literal-1. La finalización de la iteración debe indicarse por un ENDDO donde además se especifica el incremento.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

WHEN (Sentencia para condicionar ejecución de grupos de instrucciones)

La sentencia WHEN causa que una condición específica sea evaluada, agrupando varias y ejecutando solo la que sea verdadera.

Formato

WHEN { Identificador-1 } { *EQ *NE *LK } { Identificador-2 }
 { Literal-1 } { *GT *LT *NL } { Literal-2 }
 { *GE *LE }

.....

WHEN { Identificador-1 } { *EQ *NE *LK } { Identificador-2 }
 { Literal-1 } { *GT *LT *NL } { Literal-2 }
 { *GE *LE }

.....

WHEN { Identificador-1 } { *EQ *NE *LK } { Identificador-2 }
 { Literal-1 } { *GT *LT *NL } { Literal-2 }
 { *GE *LE }

.....

WHEN

ENDWH

- *El Identificador-1 o el Literal-1 es evaluado contra el Identificador-2 o Literal-2. Se ejecuta las instrucciones que solo cumplen con la condición Verdadero buscando luego el ENDWH. El WHEN sin condición se ejecuta incondicionalmente si alguno de los del grupo previo no lo hizo antes.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CASE (Sentencia para condicionar ejecución de grupos de rutinas de usuarios)

La sentencia CASE causa que una condición específica sea evaluada, agrupando varias y ejecutando solo la que sea verdadera.

Formato

```

CASE { Identificador-1 } { *EQ *NE *LK } { Identificador-2 } { Etiqueta-1 }
    { Literal-1 }      { *GT *LT *NL } { Literal-2 }      {
.....
CASE { Identificador-1 } { *EQ *NE *LK } { Identificador-2 } { Etiqueta-1 }
    { Literal-1 }      { *GT *LT *NL } { Literal-2 }      {
.....
CASE
.....

ENDCS
    
```

- *El Identificador-1 o el Literal-1 es evaluado contra el Identificador-2 o Literal-2. Se ejecuta la subrutina que se indica en ETIQUETA-1 siempre que se cumplen con la condición Verdadero, buscando luego el ENDCS. El CASE sin condición se ejecuta incondicionalmente si alguno de los del grupo previo no lo hizo antes.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

OPERACIONES DE BIFURCACIONES

GOTO (Sentencia para bifurcar a una etiqueta)

La sentencia GOTO causa que el control del programa pase a partir de una etiqueta indicada.

Formato

GOTO { Etiqueta-1 }

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al lugar donde se producirá la bifurcación.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

TAG (Sentencia para indicar una etiqueta)

La sentencia TAG define una etiqueta.

Formato

TAG { **Etiqueta-1** }

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al nombre para ser referenciada.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LOOP (Sentencia para controlar otra iteración)

La sentencia LOOP causa que el programa continúe con el lazo, realizando nueva comparación o aumentando la variable índice.

Formato

LOOP { }

➤ Se utiliza en conjunción con las sentencias *DO*, *DOUNTIL* y *DOWHILE*.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXIT (Sentencia para salir de la iteración)

La sentencia EXIT causa que el programa termine con el lazo, buscando el ENDDO, ENDDU o ENDDW.

Formato

EXIT { }

➤ Se utiliza en conjunción con las sentencias DO, DOUNTIL y DOWHILE.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

STREVN (Función para reiniciar la ejecución del evento)

La función STREVN causa que se reinicie la ejecución del evento

Formato

STREVN

➤ *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

ENDEVN (Función para finalizar la ejecución del evento)

La función ENDEVN causa que se finalice la ejecución del evento

Formato

ENDEVN

➤ *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES FECHA

WRKFEC (Función para trabajar con fechas)

La función WRKFEC toma acciones sobre una fecha dada y obtiene una nueva fecha o una forma de expresión diferente.

Formato

WRKFEC { Identificador-1 } { Literal-1 } { Literal-2 } { *D *F *G *J } { Identificador-2 }

- El Identificador-1 o Literal-1 debe contener la fecha en CCAAMMDD a tratar.
- El Literal-2 puede indicar la cantidad de días a sumar a la fecha.
- Forma en que se almacenará el resultado
 - *D calcula el día de la semana
 - *G convierte la fecha indicada a gregoriano.
 - *J convierte la fecha indicada a juliano.
 - *F almacena en CCAAMMDD
- El Identificador-2 será donde se almacenará el resultado

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CLCFEC (Función para calcular fechas)

La función CLCFEC toma acciones sobre un formato de fecha juliano o gregoriano dado y obtiene una nueva fecha en formato CCAAMMDD.

Formato

$$\text{CLCFEC} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} *G \\ *J \end{array} \right\} \left\{ \text{Identificador-2} \right\}$$

- *El Identificador-1 o Literal-1 debe contener la fecha en días Julianos o Gregorianos*
- *Tipo:*
 - *G indica que los días están expresados en gregorianos*
 - *J indica que los días están expresados en Julianos.*
- *El Identificador-2 será donde se almacenará el resultado y tendrá formato CCAAMMDD.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CLCJUL (Función para calcular días julianos o gregorianos desde fecha)

La función CLCJUL calcula días julianos o gregorianos, desde fecha en CCAAMMDD.

Formato

$$\text{CLCJUL} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} *G \\ *J \end{array} \right\} \left\{ \text{Identificador-2} \right\}$$

- *El Identificador-1 o Literal-1 debe contener la fecha en CCAAMMDD*
- *Tipo:*
 - *J indica que los días están expresados en Julianos.*
 - *G indica que los días están expresados en Gregorianos.*
- *El Identificador-2 será donde se almacenará el resultado en días julianos.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FECALF (Función para convertir una fecha en alfabética)

La función FECALF convierte una fecha de formato dd/mm/cccc en xx de mmmmmmmm de cccc.

Formato

FECALF { Identificador-1 } { Identificador-2 }
 { Literal-1 }

- *El Identificador-1 o Literal-1(numérica) debe contener la fecha.*
- *El Identificador-2 será donde se almacenará el resultado como fecha alfabética.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CLCDIA (Función para calcular días entre dos fechas)

La función CLCDIA calcula días entre dos fechas de formato CCAAMMDD.

Formato

CLCDIA { Identificador-1 } { Identificador-2 } { Identificador-3 }
 { Literal-1 } { Literal-2 } { }

- *El Identificador-1 o Literal-1 debe contener la primera fecha en CCAAMMDD a tratar.*
- *El Identificador-2 o Literal-2 debe contener la segunda fecha en CCAAMMDD a tratar.*
- *El Identificador-3 contendrá el resultado expresado en días de la diferencia entre las dos fechas.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

VALFEC (Función para validar una fecha)

La función VALFEC verifica que la fecha indicada sea válida como tal.

Formato

VALFEC { Identificador-1
Literal-1 } { Identificador-2 }

{ *V
*L } { Identificador-3
Literal-3
*NONE } { Identificador-4
Literal-4
*NONE }

- *El Identificador-1 o Literal-1 debe contener la fecha en formato CCAAMMDD.*
- *El Identificador-2 tendrá un código correspondiente al resultado de la validación (0 = sin error, 8=error de fecha invalida, 1 a 7=no laborable e indican el día de la semana) Variable donde se deja el resultado.*
- *Tipo:*
 - *V Validación de fecha*
 - *L Validación de fecha y día Laboral*
- *El Identificador-3 o Literal-3(número) debe contener código de provincia o indicar *NONE.*
- *El Identificador-4 o Literal-4(número) debe contener código de plaza o indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

VALHAB (Función para validar una fecha habil)

La función VALHAB verifica que la fecha indicada sea válida como tal y que a su vez sea hábil.

Formato

VALHAB { Identificador-1 } { Literal-1 } { Literal-2 } { Etiqueta-1 } { Identificador-3 } { Literal-3 } { Identificador-4 } { Literal-4 } { *NONE }

- El Identificador-1 o Literal-1 debe contener la fecha en formato CCAAMMDD.
- El Literal-2(número) indica el código del error. En caso ser inválida o inhábil, este código es movido a la variable &HDRTRN.
- La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.
- El Identificador-3 o Literal-3(número) debe contener código de provincia o indicar *NONE.
- El Identificador-4 o Literal-4(número) debe contener código de plaza o indicar *NONE.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXTIME (Función para recuperar fecha y hora actual)

La función EXTIME permite elegir el formato en que obtendrá la hora o la fecha del computador.

Formato

$$\text{EXTIME} \left\{ \begin{array}{l} *F \\ *H \end{array} \right\} \left\{ \begin{array}{ll} *1 & *2 \\ *1 & *2 \quad *3 \end{array} \right\} \left\{ \text{Identificador-1} \right\}$$

- **F* indica que se quiere obtener la fecha.
 - **1* en formato DDMMCCAA
 - **2* en formato CCAAMMDD
- **H* indica que se quiere obtener la hora.
 - **1* en formato HHMMSS
 - **2* en formato HHMM
 - **3* en formato HH
- El Identificador-1 almacenará el resultado.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES DE LECTURA

LEEREG (Función para acceder a un registro de un archivo)

La función LEEREG realiza la lectura de un registro, dejando disponible el contenido de todos los campo.

Formato

$$\text{LEEREG} \left\{ \begin{array}{l} \text{*Etiqueta-1} \end{array} \right\} \left\{ \begin{array}{l} \text{*Key} \quad \text{*Next} \quad \text{*NextEQ} \\ \text{*StartLT} \quad \text{*Previus} \quad \text{*PreviusEQ} \\ \text{*StartGT} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$$

- La Etiqueta-1 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al archivo sobre el cual se aplicará la función.
- Tipo:
 - *key (lectura por clave)
 - *startLT (Ubicación del puntero de lectura por menor que)
 - *startGT (Ubicación del puntero de lectura por mayor que)
 - *Next (Lectura secuencial registro siguiente)
 - *Previus (Lectura secuencial registro anterior)
 - *NextEQ(Lectura del siguiente registro de igual clave para índices duplicados)
 - *PreviusEQ(Lectura del anterior registro de igual clave para índices duplicados)
- El Identificador-1 tendrá un código correspondiente al resultado de la lectura (0=Sin error, 1=No existe registro, se llevo al fin del archivo o se llevo al inicio del archivo).Variable donde se alojará el resultado.
- El Identificador-2 o Literal-2 debe contener los valores de las claves del archivo para su lectura o ubicación del puntero.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LEECPT (Función para acceder a un registro de comprobante)

La función LEECPT realiza la lectura de un registro de comprobante.

Formato

LEECPT	{ Identificador-1 Literal-1 }	{ Literal-2 }	{ Etiqueta-1 *NONE }
	{ Identificador-3 Literal-3 }	{ Identificador-4 Literal-4 }	{ Identificador-5 Literal-5 }
	{ Identificador-6 Literal-6 }	{ Identificador-7 Literal-7 }	

- El Identificador-1 o el Literal-1 debe ser numérico e indica el prefijo que tendrá el tipo de registro (identifica un comprobante en particular), en caso de no querer utilizarlo se debe utilizar *NONE.
- *El Literal-2 (número) indica el código del error. En caso de encontrarse al registro, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 debe ser el nombre (letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*
- El Identificador-3 o el Literal-3 debe contener el Número de Sucursal.
- El Identificador-4 o el Literal-4 debe contener el Número de Cuenta.
- El Identificador-5 o el Literal-5 debe contener el Subnúmero de cuenta.
- El Identificador-6 o el Literal-6 debe contener el Número de Comprobante.
- El Identificador-7 o el Literal-7 debe contener el Subnúmero de comprobante.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LEESDO (Función para acceder a un registro de Saldo)

La función LEESDO realiza la lectura de un registro de saldo, pudiendo para algunos casos generar el alta automáticamente.

Formato

LEESDO	{ *SI	}	{ Literal-2	}	{ Etiqueta-1	}
	{ *NO				{ *NONE	
	{ Identificador-3	}	{ Identificador-4	}	{ Identificador-5	}
{ Literal-3	{ Literal-4		{ Literal-5			
{ Identificador-6	}					
{ Literal-6						

- Indica el alta automática *SI o *NO
- *El Literal-2(número) indica el código del error. En caso de encontrarse al registro, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 debe ser el nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*
- El Identificador-3 o el Literal-3 debe contener el Número de Sucursal.
- El Identificador-4 o el Literal-4 debe contener el Número de Cuenta.
- El Identificador-5 o el Literal-5 debe contener el Subnúmero de cuenta.
- El Identificador-6 o el Literal-6 debe contener el Número de Producto.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LEEGRP (Función para acceder a un grupo de registros de un archivo)

La función LEEGRP realiza la lectura de un grupo de registros para procesarlos, pudiendo especificar una rutina que se ejecutaría antes de iniciar el acceso, otra rutina que se invoca por cada registro que se lee y una tercera rutina que se invoca al finalizar.

Formato

LEEGRP	*Etiqueta-1	Identificador-1	Etiqueta-2 *NONE
	Etiqueta-3 *NONE	Etiqueta-4 *NONE	Etiqueta-5 *NONE
	Identificador-2 Literal-2	Identificador-3 Literal-3	Identificador-4 Literal-4
	Identificador-5 *NONE Literal-5	Identificador-6 *NONE Literal-6	Identificador-7 *NONE Literal-7

- La Etiqueta-1 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al archivo sobre el cual se aplicará la función.
- El Identificador-1 tendrá un código correspondiente al resultado de la lectura (0=Sin error, 1=Error).
- La Etiqueta-2 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.
- La Etiqueta-3 corresponde al nombre(letras A...Z) de subrutina de Pre-proceso, en caso de no utilizarlo se debe indicar *NONE.
- La Etiqueta-4 corresponde al nombre(letras A...Z) de subrutina de Proceso, en caso de no utilizarlo se debe indicar *NONE.
- La Etiqueta-5 corresponde al nombre(letras A...Z) de subrutina de Pos-proceso, en caso de no utilizarlo se debe indicar *NONE.
- El Identificador-2 o Literal-2, el Identificador-3 o Literal-3 y así sucesivamente, debe contener los valores que cargarán la clave del archivo para su lectura, para lo que no se utilizaría, indicar *NONE.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LEETXT (Función para leer un registro de información variable)

La función LEETXT accede a un registro de un archivo de información variable.

Formato

LEETXT { **Identificador-1** }

- *El Identificador-1 tendrá un código correspondiente al resultado de la lectura (0=Sin error, 1=Error).*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

SQLLEE (Función para leer un registro por SQL)

La función SQLLEE accede a un registro de un archivo.

Formato

SQLLEE { tabla } { Literal-1 } { Etiqueta-1 } { Literal-2 }

➤ *Tablas Utilizadas:*

*CLI	<i>Clientes</i>
*CTA	<i>Cuentas</i>
*ODP	<i>Ordenes de Pago</i>
*SDO	<i>Saldos</i>
*CPT	<i>Comprobantes</i>
*MOV	<i>Movimientos</i>
*PLF	<i>Certificados Plazo Fijo</i>
*TRN	<i>Transacciones</i>

- *El Literal-1(número) indica el código del error. En caso de encontrarse al registro, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 debe ser el nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*
- *El Literal-2(alfanumérico) debe contener las sentencias SQL*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CTLAPL (Función para controlar aplicabilidad de un importe)

La función CTLAPL controla si se puede aplicar un importe.

Formato

CTLAPL { Identificador-1 } { Literal-2 } { Etiqueta-1 }
 { Literal-1 } { *NONE }

- *El Identificador-1 o Literal-1 debe contener el importe que controla.*
- *El Literal-2(número) indica el código del error. Si no se puede aplicar el importe, se produce el error y este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CTLIMP (Función para controlar un importe)

La función CTLIMP controla un importe, el cual debe ser mayor o igual al mínimo y a su vez menor o igual al máximo especificado en el perfil del usuario.

Formato

CTLIMP	{	Identificador-1	}	{	Identificador-2	}	{	Identificador-3	}
		Literal-1			Literal-2			Literal-3	
	{	Literal-4	}	{	Etiqueta-1	}			
					*NONE				

- *El Identificador-1 o Literal-1 debe contener el importe que controla.*
- *El Identificador-2 o Literal-2 debe contener uno de los rangos, importe mínimo*
- *El Identificador-3 o Literal-3 debe contener uno de los rangos, importe máximo.*
- *El Literal-4(número) indica el código del error. Si no se puede aplicar el importe, se produce el error y este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CTLACC (Función para controlar la acción de un comprobante)

La función CTLACC controla la acción de un comprobante, previamente accedido, indicando el número de acción a controlar, en donde 1 indica ingreso y 2 egreso.

Formato

$$\text{CTLACC} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Etiqueta-1} \\ *NONE \end{array} \right\}$$

- *El Identificador-1 o Literal-1(número) debe contener el número de acción que se quiere controlar. 1= Ingreso en Caja. 2= Egreso en caja*
- *El Literal-2(número) indica el código del error. Si no es satisfactorio el control, se produce el error y este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES DE IMPRESION

OPNPRT (Función para habilitar un archivo de impresión)

La función OPNPRT establece los valores para un archivo de impresora y realiza la apertura del mismo, según la longitud definida para las líneas. Antes debe indicarse la función EXEC para especificar los atributos particulares de la impresión. EXEC OVRPRTF FILE () TOFILE (/) DEV ().

Formato

OPNPRT { Identificador-1 } { Identificador-2 } { Identificador-3
Literal-1 } { Literal-2 } { Literal-3
*NONE }

- Esta función asignará valor a dos variables *OVRFLW (control de overflow –verdadero-) y *TRANSP(control de transporte –falso-).
- El Identificador1 o el Literal1(numérico) contiene el largo de la línea a utilizar para la impresión (Long Line).
- El Identificador-2 o el Literal-2(numérico) contiene la cantidad de líneas a utilizar para la impresión(Lines).
- El Identificador-3 o el Literal-3(numérico) contiene el número de líneas a considerar para activar el overflow, en caso de querer que se calcule automáticamente se debe indicar *NONE.

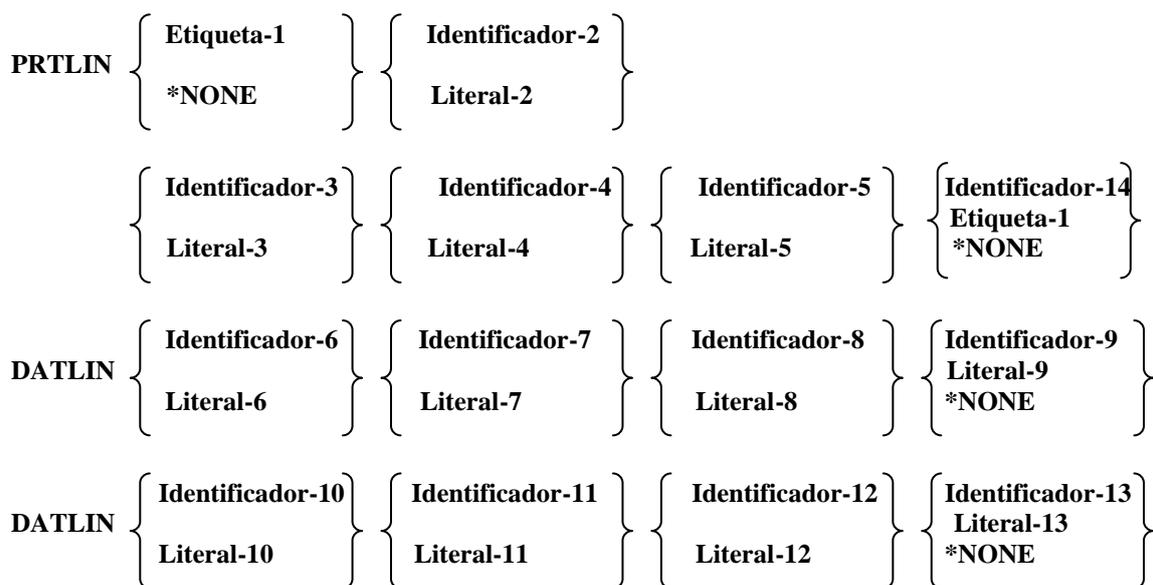
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

PRTLIN (Función para imprimir una línea con información)

La función PRTLIN permite la impresión de una línea la cual se construye con campos, indicando en cada uno de ellos la columna inicial, la longitud y el código de edición (para campos numéricos solamente). Además, controla el estado de la variable *OVRFLW, si es verdadero invocará a la rutina indicada por tal caso, la cual se encargará de imprimir los títulos del listado.

Formato



ENDPR

Por cada impresión se actualiza el número de línea activa y si corresponde se activan las variables *OVRFLW y *TRANSP

- La Etiqueta-1 corresponde al nombre(letras A...Z) de subrutina de Overflow, en caso de no utilizarlo se debe indicar *NONE
- El Identificador-2 o el Literal-2(numérico) debe contener el número de línea donde se realizará la impresión, en caso de no desear indicarlo se debe especificar 0.
- El Identificador-3 o el Literal-3(numérico) debe contener el número de línea donde se ubicará después de imprimir, en caso de no desear indicarlo se debe especificar 0.
- El Identificador-4 o el Literal-4(numérico) debe contener la cantidad de líneas que avanzará antes de imprimir, en caso de no desear indicarlo se debe especificar 0.
- El Identificador-5 o el Literal-5(numérico) debe contener la cantidad de líneas que avanzará después de imprimir, en caso de no desear indicarlo se debe especificar 0.
- El Identificador-14 o Etiqueta-1 debe contener la etiqueta del formulario definido para la impresión, en caso de no desear indicarlo se debe especificar *NONE.

DATLIN

- El Identificador-6 o el Literal-6 debe ser el nombre de campo (comienza con &) o la literal respectivamente, que se quiera imprimir.
- El Identificador-7 o el Literal-7 debe indicar el número de columna de impresión donde de colocará el campo.
- El Identificador-8 o el Literal-8 debe indicar la longitud de lo que se desea imprimir, teniendo en cuenta que es campo a campo.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

- *El Identificador-9 o el Literal-9 debe contener el código de edición a utilizar. El mismo existe en un archivo, soportandose como predefinidos a los del lenguaje RPG, con posibilidad de agregar y/o modificar otros. Para los campos que contienen datos alfanuméricos se debe indicar *NONE..*
- *El Identificador-10/11/12/13 o el Literal-10/11/12/13 deben tener igual definición que el Identificador-6/7/8/9 o el Literal-6/7/8/9, y así sucesivamente definiendo todos los campos que pertenecerían a la línea de impresión hasta finalizar con ENDPR.*

CLOPRT (Función para cerrar un archivo de impresión)

La función CLOPRT cierra el archivo de impresión correspondiente.

Formato

CLOPRT

- *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

PRTFOR (Función para imprimir un formulario)

La función PRTFOR permite la impresión de un formulario definido.

Formato

PRTFOR	{	Identificador-1	}	{	Identificador-2	}		
		Literal-1			Literal-2			
		Identificador-3		Identificador-4		*COMUN	{	Identificador-5
		Literal-3		Literal-4		*PRENUME		Literal-5
						*NUMERAR		
		Identificador-6		Identificador-7		Etiqueta-1	{	
		Literal-6		Literal-7		*NONE		
		Etiqueta-2		*SPL		Etiqueta-3	{	
		*NONE		*TXT		*NONE		
				*SPLVIEW				
				*TXTVIEW				

- El Identificador-1 o el Literal-1(alfanumerico) debe contener el código de talonario.
- El Identificador-2 o el Literal-2(merico) debe contener el código de sub-talonario
- El Identificador-3 o el Literal-3(merico) debe contener el código de sucursal.
- El Identificador-4 o el Literal-4(merico) debe contener la cantidad de copias.
- Tipo de formulario :
 - *COMUN (sin confirmar impresión)
 - *PRENUME (formulario pre-numerado)
 - *NUMERAR (formulario para numerar)
- El Identificador-5 o el Literal-5(merico) debe contener el nro. Línea para confirmar.
- El Identificador-6 o el Literal-6(merico) debe contener el nro. Columna para confirmar.
- El Identificador-7 o el Literal-7(merico) debe contener la cantidad de líneas para confirmatr.
- Etiqueta-1 debe contener el nombre de Rutina de Impresión, si no se desea indicar *NONE.
- Etiqueta-2 debe contener el nombre de etiqueta para cancelación, si no se desea indicar *NONE.
- Tipo de salida:
 - *SPL (a spool)
 - *TXT (a reporte)
 - *SPLVIEW (a spool y visualización)
 - *TXTVIEW (a reporte y visualización)
- Etiqueta-3 debe contener el nombre del reporte, si no se desea indicar *NONE (sin nombre).

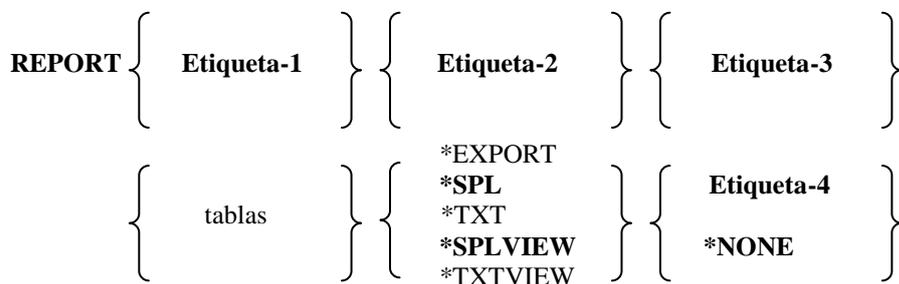
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

REPORT (Función para imprimir un reporte)

La función REPORT permite la impresión de un formulario definido.

Formato



- *Etiqueta-1 debe contener el nombre de diseño del reporte.*
- *Etiqueta-2 debe contener la rutina de selección de registros.*
- *Etiqueta-3 debe contener la rutina de proceso de registros.*
- *Tablas Utilizadas:*

*CLI	Clientes
*CTA	Cuentas
*MOV	Movimientos de Cuentas
*REL	Relación Ctas-Clientes
*TRN	Transacciones
*CPT	Comprobantes
*DOM	Domicilios
*PLF	Certificado Plazo Fijo
*SDO	Saldos Operativos

- *Tipo de salida:*

*EXPORT	(exportar datos)
*SPL	(a spool)
*TXT	(a reporte)
*SPLVIEW	(a spool y visualización)
*TXTVIEW	(a reporte y visualización)

- *Etiqueta-4 debe contener el nombre del reporte, si no se desea indicar *NONE (sin nombre).*

APLICATIVO BANCARIO GENERICO

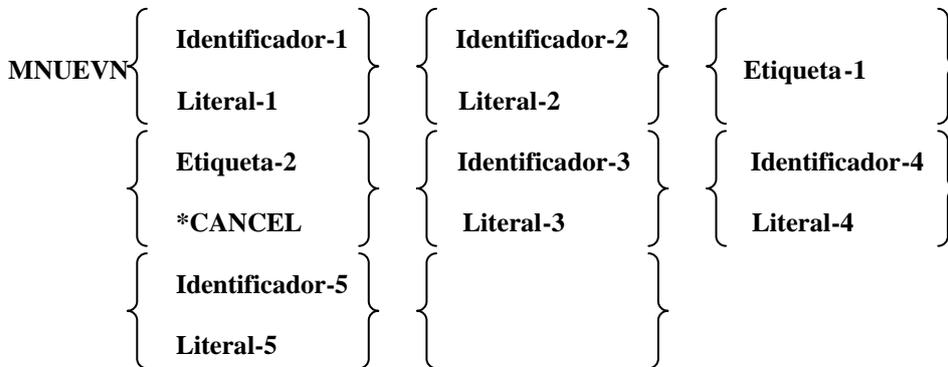
Sentencias de Codificación

FUNCIONES DE PANTALLA

MNUEVN (Función para visualizar un menú con una lista de eventos)

La función MNUEVN permite la creación de opciones dentro de un cuadro de dialogo. Se ubica cada formato determinado en la línea y columna donde se lo quiere representar.

Formato



- *El Identificador-1 o el Literal-1(numérico) debe contener el evento para la lista. Desde.*
- *El Identificador-2 o el Literal-2(numérico) debe contener el evento para la lista. Hasta.*
- *La Etiqueta-1 debe ser la identificación(letras A...Z) del formato para ser referenciado desde cancelaciones posteriores.*
- *La Etiqueta-2 debe contener la identificación de cancelación, en caso de que se quiera finalizar con este formato se debe indicar *CANCEL.*
- *El Identificador-3 o el Literal-3(numérico) debe contener el número de línea para visualizar el menú.*
- *El Identificador-4 o el Literal-4(numérico) debe contener el número de columna para visualizar el menú.*
- *Identificador-5 o Literal-5 debe contener el Título del menú*

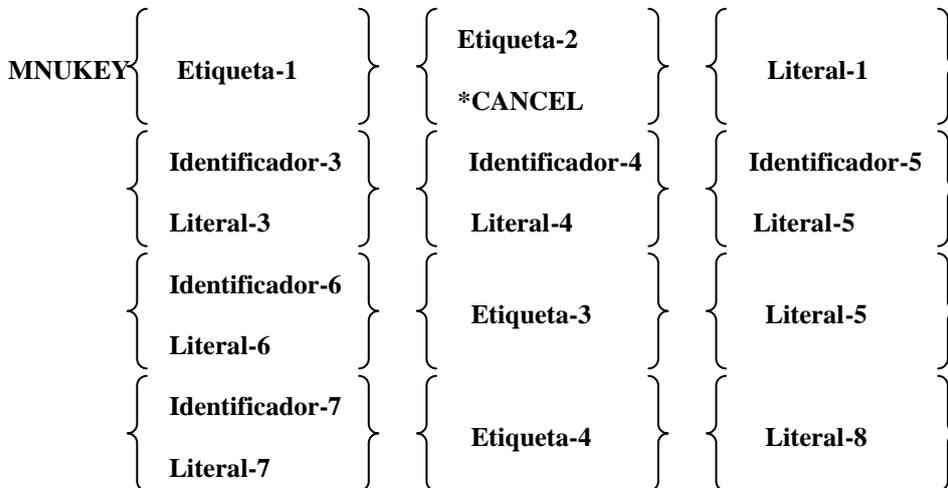
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MNUKEY (Función para visualizar un menú de teclas rápidas)

La función MNUKEY permite la creación de un menú de teclas rápidas. Se utiliza para entrelazar varios menues.

Formato



- La Etiqueta-1 debe ser la identificación(letras A...Z) del formato para ser referenciado desde cancelaciones posteriores.
- La Etiqueta-2 debe contener la identificación de cancelación, en caso de que se quiera finalizar con este formato se debe indicar *CANCEL.
- Literal-1(numérico) debe indiar el nivel del menú (de 1 a 9).
- El Identificador-3 o el Literal-3(numérico) debe contener el número de línea para visualizar el menú.
- El Identificador-4 o el Literal-4(numérico) debe contener el número de columna para visualizar el menú.
- Identificador-5 o Literal-5 debe contener el Título del menú
- Identificador-6 o Literal-6 debe contener el Texto de la opción 1
- La Etiqueta-3 debe contener la etiqueta del sub-menu 1.
- Literal-5(numérico) debe indiar el nivel mínimo para la ejecución 1.
- Identificador-7 o Literal-7 debe contener el Texto de la opción 2
- La Etiqueta-4 debe contener la etiqueta del sub-menu 2.
- Literal-8(numérico) debe indiar el nivel mínimo para la ejecución 2.

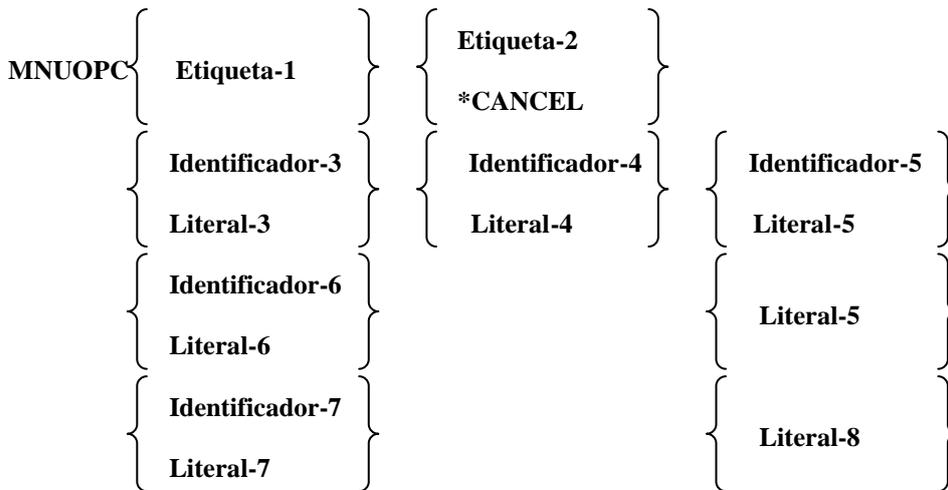
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MNUOPC (Función para visualizar un menú de opciones)

La función MNUKEY permite la creación de un menú de opciones.

Formato



- *La Etiqueta-1 debe ser la identificación(letras A...Z) del formato para ser referenciado desde cancelaciones posteriores.*
- *La Etiqueta-2 debe contener la identificación de cancelación, en caso de que se quiera finalizar con este formato se debe indicar *CANCEL.*
- *El Identificador-3 o el Literal-3(numérico) debe contener el número de línea para visualizar el menú.*
- *El Identificador-4 o el Literal-4(numérico) debe contener el número de columna para visualizar el menú.*
- *Identificador-5 o Literal-5 debe contener el Título del menú*
- *Identificador-6 o Literal-6 debe contener el Texto de la opción 1*
- *Literal-5(numérico) debe indicar el nivel mínimo para la ejecución 1.*
- *Identificador-7 o Literal-7 debe contener el Texto de la opción 2*
- *Literal-8(numérico) debe indicar el nivel mínimo para la ejecución 2.*

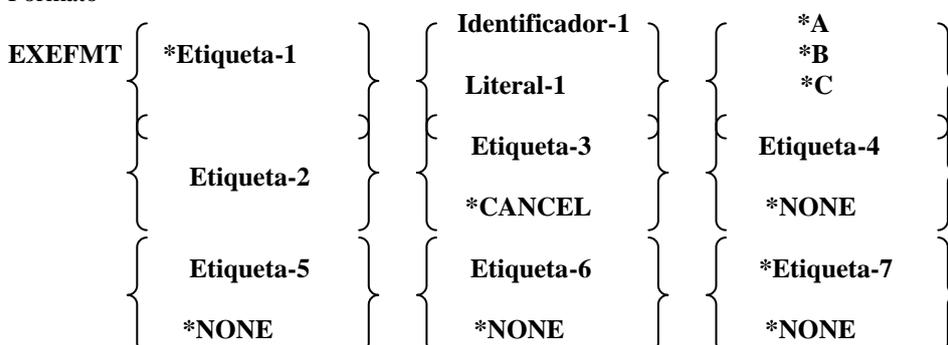
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXEFMT (Función para visualizar e ingresar datos por pantalla)

La función EXEFMT permite visualizar e ingresar datos por pantalla. Se ubica cada formato determinado en la línea y cuadrante donde se lo quiere representar. Hay que tener en cuenta que cada formato ocupa tres líneas y que la pantalla está separada en tres columnas (cuadrante A, B y C). Las líneas disponibles son de la número 07 a la 20 inclusive, las restantes están reservadas para otros efectos.

Formato



- La Etiqueta-1 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al formato sobre el cual se aplicará la función. Los formatos disponibles son:
 - *CTATRN (Cuenta transacción. Solicita. Sucursal, Cuenta y Subnúmero). Carga las variables XOSUCU, XONCTA y XOSNUM. Ocupa tres cuadrantes por lo que en el Literal-3 se debe indicar A o dejar que asuma su defecto en cual es A.
 - *CTAREF (Cuentas Referenciadas. Solicita Sucursal, Cuenta, Subnúmero y Producto). Carga las variables XORSUC, XORCTA, XORSUB y XOPROD. Ocupa tres cuadrantes por lo que en el Literal-3 se debe indicar A o dejar que asuma su defecto en cual es A.
 - *INTCTA(Integrantes de la cuenta). Muestra un subfile con los integrantes.
 - *DATCLI(Permite el alta o modificación de los datos del cliente). Muestra un subfile con los datos.
 - *FECHA1 (Ingresa una fecha DDMCCAA). Carga la variable XOFEC1 la que se almacena como CCAAMMDD, previamente es validada según las consideraciones para campos fecha.
 - *FECHA2 (Ingresa una fecha DDMCCAA). Carga la variable XOFEC2 la que se almacena como CCAAMMDD, previamente es validada según las consideraciones para campos fecha.
 - *FECHA3 (Ingresa una fecha DDMCCAA). Carga la variable XOFEC3 la que se almacena como CCAAMMDD, previamente es validada según las consideraciones para campos fecha.
 - *CPT1 (Ingresa número de comprobante de la transacción). Carga la variable XONCPT.
 - *CPT2 (Ingresa número de comprobante de referencia de la transacción y subcomprobante). Carga la variable XONREF y XOSREF
 - *CPT3 (Ingresa datos de referencia adicional o una observación). Carga la variable XODREF.
 - *IMP1 (Ingresa y visualiza importe). Carga la variable XOIMP1.
 - *IMP2 (Ingresa y visualiza importe). Carga la variable XOIMP2.
 - *IMP3 (Ingresa y visualiza importe). Carga la variable XOIMP3.
 - *CANT(Ingresa y visualiza cantidad). Carga la variable XOCANT.
- El Identificador-1 o el Literal-1(numérico) debe contener el número de línea donde de ubicará el formato (Línea 07, 10, 13, 16, o 19).
- *A, *B o *C y corresponde al cuadrante donde se ubicará el formato.
- La Etiqueta-2 debe ser la identificación(letras A...Z) del formato para ser referenciado desde cancelaciones posteriores.
- La Etiqueta-3 debe contener la identificación de cancelación, en caso de que se quiera finalizar con este formato se debe indicar *CANCEL.

APLICATIVO BANCARIO GENERICO

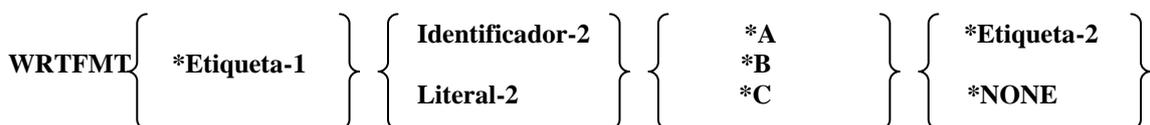
Sentencias de Codificación

- La Etiqueta-4 corresponde al nombre(letras A...Z) de subrutina de Pre-proceso, en caso de no utilizarlo se debe indicar *NONE.
- La Etiqueta-5 corresponde al nombre(letras A...Z) de subrutina de Validación, en caso de no utilizarlo se debe indicar *NONE.
- La Etiqueta-6 corresponde al nombre(letras A...Z) de subrutina de Proceso o indicar *NONE.
- La Etiqueta-7 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al nombre del texto de ayuda operativo. En caso de no definirlo se debe indicar *NONE..

WRTFMT (Función para visualizar datos por pantalla)

La función WRTFMT permite visualizar datos por pantalla. Se ubica cada formato determinado en la línea y cuadrante donde se lo quiere representar. Hay que tener en cuenta que cada formato ocupa tres líneas y que la pantalla está separada en tres columnas (cuadrante A, B y C). Las líneas disponibles son de la número 07 a la 20 inclusive, las restantes están reservadas para otros efectos.

Formato



- La Etiqueta-1 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al formato sobre el cual se aplicará la función. Los formatos disponibles son:
 - *CTATRN (Cuenta transacción. Visualiza Sucursal, Cuenta y Subnúmero). Muestra las variables X0SUCU, X0NCTA y X0SNUM. Ocupa tres cuadrantes por lo que en el Literal-3 se debe indicar A.
 - *CTAREF (Cuenta Referencia. Solicita Sucursal, Cuenta, Subnúmero y Producto). Muestra las variables X0RSUC, X0RCTA, X0RSUB y X0PROD. Ocupa tres cuadrantes por lo que en el Literal-3 se debe indicar A.
 - *INTCTA(Integrantes de la cuenta). Muestra un subfile con los integrantes.
 - *DATCLI(Permite el alta o modificación de los datos del cliente). Muestra un subfile con los datos.
 - *FECHA1(Visualiza una fecha DDMCCAA). Muestra la variable X0FEC1.
 - *FECHA2(Visualiza una fecha DDMCCAA). Muestra la variable X0FEC2.
 - *FECHA3(Visualiza una fecha DDMCCAA). Muestra la variable X0FEC3.
 - *CPT1 (Visualiza el número de comprobante de la transacción). Muestra la variable X0NCPT.
 - *CPT2 (Visualiza número de comprobante de referencia de la transacción y subcomprobante). Muestra la variable X0NREF y X0SREF
 - *CPT3 (Visualiza datos de referencia adicional o una observación). Muestra la variable X0DREF.
 - *IMP1 (Visualiza importe). Muestra la variable X0IMP1.
 - *IMP2 (Visualiza importe). Muestra la variable X0IMP2.
 - *IMP3 (Visualiza importe). Muestra la variable X0IMP3.
 - *CANT(Visualiza cantidad). Muestra la variable X0CANT.
- El Identificador-2 o el Literal-2(numérico) debe contener el número de línea donde se ubicará el formato (Línea 07, 10, 13, 16, o 19).
- Cuadrante:
 - *A, *B o *C y corresponde al cuadrante donde se ubicará el formato.
- La Etiqueta-2 debe contener caracteres alfabéticos comenzando con letras(A...Z) y corresponde al nombre del texto de ayuda operativo. En caso de no definirlo se debe indicar *NONE..

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

INZSFL (Inicializar un Sub File de consulta)

La función INZSFL borra el contenido del subfile armado.

Formato

INZSFL

➤ *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

LOADSF (Función para cargar datos en el Sub File de consulta)

La función LOADSF permite la carga de datos de una línea la cual se construye con campos, indicando en cada uno de ellos la columna inicial, la longitud y el código de edición (para campos numéricos solamente).

Formato

LOADSF { Identificador-1 } { Identificador-2
Literal-2 } { Identificador-3
Literal-3 } { Identificador-4
Literal-4
*NONE }

DATLIN { Identificador-5 } { Identificador-6
Literal-6 } { Identificador-7
Literal-7 } { Identificador-8
Literal-8
*NONE }

DATLIN { } {
..... } {
..... } {
.....
*NONE }

ENDSFL

- El Identificador-1 debe ser el nombre de campo (comienza con &) que se quiera consultar.
- El Identificador-2 o el Literal-2 debe indicar el número de columna de la consulta donde de colocará el campo.
- El Identificador-3 o el Literal-3 debe indicar la longitud de lo que se desea consultar, teniendo en cuenta que es campo a campo.
- El Identificador-4 o el Literal-4(alfabético) debe contener el código de edición a utilizar. El mismo existe en un archivo, soportándose como pre-definidos a los del lenguaje RPG, con posibilidad de agregar y/o modificar otros. Para los campos que contienen datos alfanuméricos se debe indicar *NONE..
- El Identificador-5/6/7/8 o el Literal-6/7/8 deben tener igual definición que el Identificador-1/2/3/4 o el Literal-2/3/4, y así sucesivamente definiendo todos los campos que pertenecerían a la línea de consulta hasta finalizar con ENDSFL.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES DE VARIABLES

CONCAT (Función para concatenar dos variables en una tercera).

La función CONCAT causa que dos literales se concatenen, indicándole, a la primera, el largo que se debe considerar y a la segunda la cantidad de espacios que debe dejar antes de comenzar a unirse.

Formato

$$\text{CONCAT} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-3} \\ \text{Literal-3} \end{array} \right\}$$
$$\left\{ \begin{array}{l} *N \\ *P \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-5} \end{array} \right\}$$

- El Identificador-1 o Literal-1 debe contener la primera información para concatenar.
- El Identificador-2 o Literal-2 debe contener la segunda información para concatenar.
- El Identificador-3 o Literal-3 debe contener el número que indica cuantos espacios en blanco se dejarán después de la primera variable para comenzar con la segunda. Indicar 0 para que no se dejen espacios.
- Modo:
 - *N Normal (completa con espacios en blanco)
 - *P Porción (solo lleva a la variable de resultado lo concatenado, de tal forma que si la variable tiene información anterior solapa solo la concatenación dejando lo que no llega a cubrir.
- El identificador-5 almacenará el resultado.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

SUBSTR (Función para obtener una substring de un variable).

La función SUBSTR extrae una porción de caracteres desde una posición determinada y por una largo especificado.

Formato

$$\text{SUBSTR} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-3} \\ \text{Literal-3} \end{array} \right\}$$
$$\left\{ \begin{array}{l} *N \\ *P \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-5} \end{array} \right\}$$

- *El Identificador-1 debe contener la información con la que se quiere trabajar.*
- *El Identificador-2 o Literal-2 debe contener el valor que indica desde que posición se quiere extraer los datos.*
- *El Identificador-3 o Literal-3 debe contener el valor que indica el largo que se quiere extraer.*
- *Modo:*
 - *N Normal (completa con espacios en blanco)*
 - *P Porción (solo lleva a la variable de resultado lo extraído, de tal forma que si la variable tiene información anterior solapa solo lo extraído dejando lo que no llega a cubrir.*
- *El identificador-5 almacenará el resultado.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CVTDEC (Función para obtener campo número desde variable alfabética)

La función CVTDEC extrae una porción de caracteres desde una posición determinada y por una largo especificado y lo convierte a número

Formato

CVTDEC { Identificador-1 } { Identificador-2
Literal-2 } { Identificador-3
Literal-3 } { Identificador-4 }

- *El Identificador-1 debe contener la información con la que se quiere trabajar.*
- *El Identificador-2 o Literal-2 debe contener el valor que indica desde que posición se quiere extraer los datos.*
- *El Identificador-3 o Literal-3 debe contener el valor que indica el largo que se quiere extraer.*
- *El identificador-4 almacenará el resultado como número.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EDTDEC (Función para editar un campo numérico en una variable alfabética)

La función EDTDEC extrae una porción de números considerando desde derecha hacia izquierda, según el largo especificado y lo edita con separadores.

Formato

$$\text{EDTDEC} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-3} \\ \text{Literal-3} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-4} \end{array} \right\}$$

- *El Identificador-1 debe contener la información con la que se quiere trabajar.*
- *El Identificador-2 o Literal-2 debe contener el valor que indica la cantidad de dígitos a considerar*
- *El Identificador-3 o Literal-3 debe contener el formato de la edición encerrado entre doble comillas .
Ejem: “ . , ” o “ / / ”*
- *El identificador-4 almacenará el resultado en formato alfabético.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

ASGVAL (Función para asignar valor carácter a una variable).

La función ASGVAL asigna a una variable alfanumérica una cadena de caracteres dentro de la cual se pueden especificar variables cuyo valor serian insertados en la asignación.

Formato

$$\text{ASGVAL} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$$

- *El Identificador-1 corresponde a la variable ha asignar valor.*
- *El Identificador-2 o Literal-2 debe contener caracteres alfanuméricos.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

RQRSTR (Función para iniciar un paso de proceso)

La función RQRSTR permite iniciar un paso de proceso.

Formato

RQRSTR { Identificador-1 } { Literal-2 } { Etiqueta-1 }
 { Literal-1 } { *NONE }

- *El Identificador-1 o Literal-1 debe contener el nombre que identifica al trabajo*
- *El Literal-2(número) indica el código del error. Si el trabajo no está planificado, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXSR (Sentencia para ejecutar rutina definida por el usuario).

La sentencia EXSR causa que el programa continúe en ejecución en una subrutina y una vez finalizado regrese al paso siguiente al de esta sentencia.

Formato

EXSR { **Etiqueta-1** }

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al nombre de la subrutina que contiene las instrucciones.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

BEGSR (Sentencia para definir rutina de usuario)

La sentencia **BEGSR** es el indicador de comienzo de definición de una subrutina de usuario.

Formato

BEGSR { **Etiqueta-1** }
.....

ENDSR

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al nombre de la subrutina que previamente fue invocada. Para indicar la finalización de la definición de sentencias y/o funciones se debe expresar **ENDSR**.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXFUN (Sentencia para ejecutar función definida por el usuario).

La sentencia EXFUN causa que el programa continúe la ejecución en una función y una vez finalizado regrese al paso siguiente al de esta sentencia. Se debe tener en cuenta el informar con que datos trabajará la función, en el caso de que la misma utilice parámetros.

Formato

EXFUN { **Etiqueta-1** }

{ **Identificador-1** } { **Identificador-2** } { **Identificador-3** }

{ **Literal-1** } { **Literal-2** } { **Literal-3** }

{ **Identificador-4** } { **Identificador-5** } { **Identificador-6** }

{ **Literal-4** } { **Literal-5** } { **Literal-6** }

{ **Identificador-7** } { **Identificador-8** } { **Identificador-9** }

{ **Literal-7** } { **Literal-8** } { **Literal-9** }

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al nombre de la función que contiene la codificación.*
- *El Identificador-1 al Identificador-9 o Literal-1 al Literal-9, corresponden a los datos a emplear y deben responder posicionalmente a los empleados en la codificación de la función que invoca.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

BEGFU (Sentencia para definir función de usuario).

La sentencia BEGFU es el indicador de comienzo de definición de una función de usuario.

Formato

BEGFU { **Etiqueta-1** }

{ **Identificador-1** } { **Identificador-2** } { **Identificador-3** }

{ **Identificador-4** } { **Identificador-5** } { **Identificador-6** }

{ **Identificador-7** } { **Identificador-8** } { **Identificador-9** }

{ *Codificación de la función* }

.....

ENDFU

- *La Etiqueta-1 debe ser Alfanumérico comenzando con letras(A...Z) y corresponde al nombre de la función que previamente fue invocada. Para indicar la finalización de la definición de sentencias y/o subrutinas se debe expresar ENDFU.*
- *El Identificador-1 al Identificador-9 (deben comenzar con %) y, corresponden a los datos a emplear y deben responder posicionalmente a los empleados en la invocación de esta función.*
- *En la codificación se deben emplear las variables definidas dentro de los identificadores.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CALL (Función para invocar a otro evento)

La función CALL produce una llamada a otro evento de las misma transacción o de otra especificada.

Formato

$$\text{CALL} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\}$$

- *El Identificador-1 o Literal-1 debe contener número de transacción*
- *El Identificador-2 o Literal-2 debe contener número del evento.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

RETURN (Función para finalizar la ejecución de sentencias)

La función RETURN produce la finalización del evento.

Formato

RETURN

➤ *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXEC (Función para invocar a un comando del sistema operativo)

La función EXEC produce una llamada al sistema operativo para la ejecución de comandos.

Formato

$$\text{EXEC} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\}$$

- *El Identificador-1 o Literal-1 debe contener el comando a ejecutar. Para indicar continuación al finalizar la línea de codificación se expresa + y si se desea utilizar variables referenciales se deben antemponer el carácter &. Todo debe ir encerrado entre " (comillas dobles).*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

SNDMSG (Función para enviar mensajes al cliente)

La función SNDMSG permite “conversar” entre aplicativos de distintas plataformas.

Formato

$$\text{SNDMSG} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Literal-3} \end{array} \right\}$$

- *El Identificador-1 o Literal-1(número) corresponde a la variable de error.*
- *El Literal-2(número) corresponde a la información de coordinación, generalmente se indica 0.*
- *El Literal-3(número) corresponde a la cantidad de datos que acompañaran al mensaje.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

WRMSG (Función para visualizar mensajes y continuar)

La función WRMSG permite dejar “pegado” el mensaje y continuar con la ejecución.

Formato

WRMSG	{	Identificador-1	}	{	Identificador-2	}
		Literal-1	}		Literal-2	}
		Identificador-3	}		Identificador-4	}
		Literal-3	}		Literal-4	}

- *El Identificador-1 o Literal-1(alfanumérico) corresponde a la información a mostrar. El largo máximo permitido es de 60 caracteres.*
- *El Identificador-2 o Literal-2(número) corresponde a la línea donde se mostrará el mensaje.*
- *El Identificador-3 o Literal-3(número) corresponde a la columna donde se mostrará el mensaje.*
- *El Identificador-4 o Literal-4(número) corresponde a la cantidad de líneas.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

EXEMSG (Función para visualizar mensajes y esperar)

La función EXEMSG permite visualizar un mensaje espera una acción, donde INTRO confirma y F12 cancela.

Formato

EXEMSG { Identificador-1 } { Identificador-2 } { Identificador-3 }
 { Literal-1 } { Literal-2 } { Literal-3 }

 { Identificador-4 } { Etiqueta-1 }
 { Literal-4 } *NONE
 *CANCEL

- El Identificador-1 o Literal-1(alfanumérico) corresponde a la información a mostrar. El largo máximo permitido es de 60 caracteres.
- El Identificador-2 o Literal-2(número) corresponde a la línea donde se mostrará el mensaje.
- El Identificador-3 o Literal-3(número) corresponde a la columna donde se mostrará el mensaje.
- El Identificador-4 o Literal-4(número) corresponde a la cantidad de líneas.
- La Etiqueta-1 debe ser el nombre(letras A...Z) de etiqueta donde puede bifurcar por F12, en caso de no utilizarlo se debe indicar *NONE o *CANCEL (finaliza con este formato).

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

STRDBG(Función para iniciar depuración)

La función STRDBG inicia una “TRACE” de las instrucciones ejecutadas.

Formato

STRDBG

.....

ENDDBG

➤ *No se indican parámetros. Para indicar la finalización de la depuración digitar ENDDBG.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES DE ARCHIVO TIPO TEXTO

OPNTEXT (Función para abrir un archivo de información variable)

La función OPNTEXT realiza la apertura del archivo texto (información variable). Esta función debe estar precedida por la función NMWORK la cual asigna un nombre aleatorio, que será utilizado como identificador del archivo. Si el archivo es utilizado como entrada, debe especificarse la función EXEC para realizar la copia del documento al archivo texto.

```
CPYFRMPCD FROMFLR ( ) FROMDOC (archivo.xxx) TOFILE ( \ txt)
          TOMBR (&NMWORK) REPLACE (*YES) TRNTBL (*NONE)
```

Formato

OPNTEXT { Identificador-1 } { *INPUT
*OUTPUT }

- *El Identificador-1 es la variable (comienza con &) donde está almacenada el nombre.*
- *Modo de acceso: *INPUT (de entrada) *OUTPUT (de salida)*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

CLOTXT (Función para cerrar un archivo de información variable)

La función CLOTXT cierra el archivo correspondiente. Si el archivo fue utilizado como salida de información a continuación debe ejecutarse la función EXEC.

```
CPYTOPCD FROMFILE ( \txt) FROMMBR (&NMWORK) TOFLR ( ) TODOC ( )  
REPLACE (*YES) TRNTBL (*NONE)
```

Formato

CLOTXT { **Identificador-1** }

- *El Identificador-1 tendrá un código correspondiente al resultado de la operación (0=Sin error, 1=Error).*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DEFREG (Función para definir registro de información variable)

La función DEFREG crea la estructura del registro de información variable.

Formato

DEFREG { Identificador-1
Literal-1
*NONE } { Literal-2 } { Literal-3 }

- El Identificador-1 o el Literal-1 (alfabético) debe contener el prefijo que identificará al tipo de registro, en caso de no querer definirlo se debe indicar *NONE.
- El Literal-2(alfabético) es un delimitador de campos y solo están permitidos: “; ;b *T *N (donde b es blanco, *T es TAB y *N es NONE).
- El Literal-3(alfabético) es el que identifica el tipo de separador decimal, solo están permitidos: . , (punto o coma).

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

DEFTXT (Función para redefinir posiciones de los campos predefinidos)

La función DEFTXT redefine posiciones de los campos pre-definidos de un archivo de texto

Formato

$$\text{DEFTXT} \left\{ \begin{array}{l} \text{Identificador-1} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-2} \\ \text{Literal-2} \end{array} \right\} \left\{ \begin{array}{l} \text{Identificador-3} \\ \text{Literal-3} \end{array} \right\}$$

- *El Identificador-1 debe contener el nombre que identificará al campo.*
- *El Identificador-2 o Literal-2(número) debe contener el valor que indica desde que posición se quiere definir el campo.*
- *El Identificador-3 o Literal-3(número) debe contener el valor que indica el largo que se quiere considerar.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MOVDAT (Función para mover datos al registro de información variable)

La función MOVDAT permite la carga de datos en un registro el cual se construye con campos, indicando en cada uno de ellos la columna inicial, la longitud y el código de edición (para campos numéricos solamente).

Formato

MOVDAT { Identificador-1 } { Identificador-2
Literal-2 } { Identificador-3
Literal-3 } { Identificador-4
Literal-4
*NONE }

Mover una variable al área de registro variable o al área de parámetros de salida.

- *El Identificador-1 debe ser el nombre de campo (comienza con &) que se quiera mover.*
- *El Identificador-2 o el Literal-2 debe indicar el número de columna donde de colocará el campo.*
- *El Identificador-3 o el Literal-3 debe indicar la longitud de lo que se desea mover.*
- *El Identificador-4 o el Literal-4(alfabético) debe contener el código de edición a utilizar. El mismo existe en un archivo, soportándose como pre-definidos a los del lenguaje RPG, con posibilidad de agregar y/o modificar otros. Para los campos que contienen datos alfanuméricos se debe indicar *NONE..*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

FUNCIONES INCORPORADAS

NMWORK (Función para generar un nombre de trabajo aleatorio)

La función NMWORK genera un nombre aleatorio para ser utilizado como miembro de dato de archivo a importar o exportar.

Formato

NMWORK { Identificador-1 }

- *El Identificador-1 es la variable (comienza con &) donde la función almacenará el nombre que obtuvo.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

IMPLET (Función para convertir un importe en letras)

La función IMPLET se la puede indicar como de monto escrito, es decir, convierte un importe a letras. La función tiene la posibilidad de dejar el resultado en una, dos o tres variables dependiendo de la necesidad del usuario, para lo cual se debe especificar el largo de cada una.

Formato

IMPLET { Identificador-1 } { Identificador-2 } { Identificador-3 }
 { Literal-1 } { Literal-2 } { Literal-3 }

 { Identificador-4 } { Identificador-5 }
 { Literal-5 }

 { Identificador-6 } { Identificador-7 } { Identificador-8 } { Identificador-9 }
 { *NONE } { Literal-7 } { *NONE } { Literal-9 }
 { *NONE } { *NONE }

- *Identificador-1 o Literal-1(numérico) debe contener el importe a convertir en letras*
- *Identificador-2 o Literal-2(alfabetico) debe contener el texto de prefijo (antes de la leyenda de monto escrito)*
- *Identificador-3 o Literal-3(alfabetico) debe contener el texto para la forma decimal. Ej.(“/100”).*
- *Identificador-4 nombre de la variable donde residirá(parte o totalmente) el importe en letras.*
- *Identificador-5 o Literal-5(numérico) debe indicar el largo a considerar para la conversión en la variable.*
- *Identificador-6 nombre de la segunda variable donde residirá(parte o resto) el importe en letras, en caso de no utilizar indicar *NONE.*
- *Identificador-7 o Literal-7(numérico) debe indicar el largo a considerar para la conversión en la variable, en caso de no utilizar indicar *NONE.*
- *Identificador-8 nombre de la segunda variable donde residirá(parte o resto) el importe en letras, en caso de no utilizar indicar *NONE.*
- *Identificador-9 o Literal-7(numérico) debe indicar el largo a considerar para la conversión en la variable, en caso de no utilizar indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

MOVCHQ (Función para mover un ítem del vector de cheques)

La función MOVCHQ mueve un ítem del vector de cheques. Este vector corresponde al buffer del sistema MOSAIC, por lo que los cheques(hasta 7 como máximo) son los ingresados por la caja en una operatoria específica

Formato

MOVCHQ { Identificador-1 }
 { Literal-1 }

➤ *El Identificador-1 o Literal-1(número) indica el número de ítem a mover.*

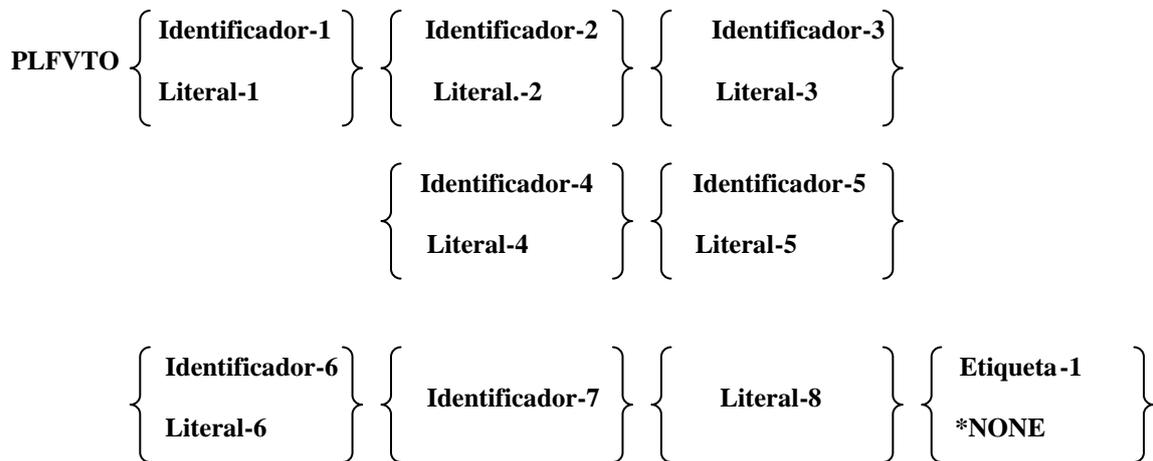
APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

PLFVTO (Función para validar fecha de vencimiento y plazo de Plazo Fijo)

La función PLFVTO determina la fecha de vencimiento del plazo fijo, indicando si la expresada es válida y/o hábil. La condición de cálculo toma en cuenta 1ro. si tiene indicado un plazo y a partir de este determina la fecha validándola luego; si no tiene indicado el plazo toma la fecha ingresada como fecha de vencimiento procediendo con esta a validarla.

Formato



- *Identificador-1 o Literal-1(numérico) debe contener la fecha de proceso. La fecha de proceso del computador está disponible en la variable &*FECPRO.*
- *Identificador-2 o Literal-2(numérico) debe contener el código de tasa.*
- *Identificador-3 o Literal-3(numérico) debe contener el sub-código de tasa.*
- *Identificador-4 o Literal-4(numérico) debe contener la fecha de la tasa que se quiere recuperar*
- *Identificador-5 o Literal-5(numérico) debe contener el monto*
- *Identificador-6 o Literal-6(numérico) debe contener el plazo.*
- *Identificador-7 variable que almacenará el resultado en forma numérica*
- *El Literal-8(número) indica el código del error. Si no existe la tasa, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

WRTTXT (Función para grabar el registro de un archivo de información variable)

La función WRTTXT realiza la grabación de un registro.

Formato

WRTTXT { Identificador-1 }

- *El Identificador-1 tendrá un código correspondiente al resultado de la grabación (0=Sin error, 1=Error).*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

GRAMOV (Función que graba un movimiento)

La función GRAMOV graba un movimiento.

Formato

GRAMOV	{	Literal-1	}	{	*N	}	{	*AFTOPE	}	{	*NONE	}
					*R							
		Identificador-1			Identificador-2							
		Literal-1			Literal-2							
		Identificador-3		Identificador-4		Identificador-5		Identificador-6				
		Literal-3		Literal-4		Literal-5		Literal-6				
		Identificador-7		Identificador-8		Identificador-9		Identificador-10				
		Literal-7		Literal-8		Literal-9		Literal-10				
		Identificador-11		Identificador-12		Identificador-13		Identificador-14				
		Literal-11		Literal-12		Literal-13		Literal-14				

- *Literal-1 debe indicar el código de movimiento ya establecido.*
- *La acción de la grabación puede ser: * N normal * R reversar*
- *El saldo operativo puede afectarse para lo cual se indica *AFTOPE o *NONE para no afectar.*
- *El Identificador-1 o Literal-1(número) debe ser fecha y debe corresponder a la fecha de proceso.*
- *El Identificador-2 o Literal-2(número) debe ser fecha y debe corresponder a la fecha de vencimiento.*
- *El Identificador-3 o Literal-3(número) debe ser fecha y debe corresponder a la fecha de gestión.*
- *El Identificador-4 o Literal-4(número) debe indicar una cantidad.*
- *El Identificador-5 o Literal-5(número) debe indicar el número principal del comprobante tratado.*
- *El Identificador-6 o Literal-6(número) debe indicar el nro de referencia del comprobante tratado.*
- *El Identificador-7 o Literal-7(número) debe indicar el nro de sub-referencia del comprob. tratado.*
- *El Identificador-8 o Literal-8 debe indicar datos u observaciones.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

- *El Identificador-9 o Literal-9(número) debe indicar el importe principal del movimiento.*
- *El Identificador-10 o Literal-10(número) debe indicar el importe de referencia del movimiento.*
- *El Identificador-11 o Literal-11(número) debe indicar un importe adicional o cero.*
- *El Identificador-12 o Literal-12(número) debe indicar la tasa 1.*
- *El Identificador-13 o Literal-13(número) debe indicar la tasa 2.*
- *El Identificador-14 o Literal-14(número) debe indicar la tasa 3*

GRACPT (Función que graba un comprobante)

La función GRACPT graba un comprobante desde los datos de la transacción.

Formato

GRACPT	Identificador-1 Literal-1	Identificador-2 Literal-2	Identificador-3 Literal-3	Identificador-4 Literal-4
	Identificador-5 Literal-5	Identificador-6 Literal-6	*ALTA *BAJA	Identificador-8 Literal-8
	Identificador-9 Literal-9	Identificador-10 Literal-10	Literal-11	Etiqueta-1 *NONE

- *El Identificador-1 o Literal-1(número) debe ser el prefijo del comprobante.*
- *El Identificador-2 o Literal-2(número) debe ser el número del comprobante.*
- *El Identificador-3 o Literal-3(número) debe ser sub-número del comprobante.*
- *El Identificador-4 o Literal-4(número) debe indicar el código de sucursal.*
- *El Identificador-5 o Literal-5(número) debe indicar el número de cuenta.*
- *El Identificador-6 o Literal-6(número) debe indicar el nro sub-número de cuenta.*
- *Tipo de acción: *ALTA *BAJA*
- *El Identificador-8 o Literal-8 debe indicar la denominación del comprobante.*
- *El Identificador-9 o Literal-9(número) debe indicar la posición frente al IVA.*
- *El Identificador-10 o Literal-10(número) debe indicar el número de CUIT.*
- *El Literal-11(número) indica el código del error. Si se produce un error en la grabación, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

GRAPLF (Función que graba un certificado de plazo fijo)

La función GRAPLF graba un certificado de plazo fijo desde los datos de la transacción.

Formato

GRAPLF { Identificador-1 } { Identificador-2 } { Literal-3 } { Etiqueta-1 }
 { Literal-1 } { Literal-2 } { *NONE }

- *El Identificador-1 o Literal-1(número) debe ser el tipo de certificado.*
- *El Identificador-2 o Literal-2(número) debe ser el tipo de garantía.*
- *El Literal-3(número) indica el código del error. Si se produce un error en la grabación, este código es movido a la variable &HDRTRN.*
- *La Etiqueta-1 corresponde al nombre(letras A...Z) de etiqueta donde puede bifurcar por error, en caso de no utilizarlo se debe indicar *NONE.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

APLIMP (Función para aplicar un importe al saldo)

La función APLIMP aplica un importe al saldo operativo y mediante una condición al saldo contable.

Formato

$$\text{APLIMP} \left\{ \begin{array}{l} \text{Identificador-1} \\ \text{Literal-1} \end{array} \right\} \left\{ \begin{array}{l} *ADD \\ *SUB \end{array} \right\} \left\{ \begin{array}{l} *SDOOPE \\ *NONE \end{array} \right\} \left\{ \begin{array}{l} *SDOCON \\ *NONE \end{array} \right\}$$

- *El Identificador-1 o Literal-1 debe contener el importe que se quiere aplicar.*
- *Tipo*
**ADD o *SUB condiciona la operación como incremental o decremental respectivamente.*
- *El valor *SDOOPE debe considerarse para la afectación sobre el saldo operativo. En caso de no querer afectarlo se debe indicar *NONE*
- *El valor *SDOCON debe considerarse para la afectación sobre el saldo contable. En caso de no querer afectarlo se debe indicar *NONE*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

COMMIT (Función para definir una unidad de procesamiento)

La función COMMIT inicia una unidad de procesamiento

Formato

COMMIT

ENDCOMMIT

- *No se indican parámetros.*
- *Para la finalización indicar ENDCOMMIT.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

ROLLBACK (Función para anular operaciones)

La función ROLLBACK anula las operaciones efectuadas hasta el COMMIT anterior.

Formato

ROLLBACK

➤ *No se indican parámetros.*

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

NOMECLATURA y DEFINICIONES

Comentario dentro de sentencias (/ * comentario *)

- Corresponden a caracteres que deben ir precedidos por / * y finalizan con */.

Literal Numérica (0/9, -, +)

- Corresponde a valores que deben estar expresados con dígitos que van del 0 al 9 inclusive. Permite una variación para indicar valores positivos o negativos, para lo cual se debe anteponer + o - respectivamente. Su máxima definición es de 16 enteros y nueve decimales.

Literal Alfanumérica (“caracteres”)

- Corresponde a caracteres que deben estar encerrados por *doble comilla* (“), siendo su máxima longitud de 24.

Campos de tablas y Variables (&campo, %campo)

- Dentro de esta categoría existen dos tipos:
 - Campos o variables internas definidas
 - Deben ir precedidas por el signo *apersen* (&)
 - Campos o variables internas utilizadas como parámetros
 - Deben ir precedidas por el signo *por ciento* (%)

Palabras Reservadas (*AAAAAAAAA)

- Corresponden a caracteres que van precedidos por un *asterisco* (*)

Variables de uso general

- *ZEROES (ceros)
- *BLANKS (blancos)
- *HIVAL (máximo valor)
- *LOVAL (mínimo valor)
- *ALL 'x' (llenar todo con los caracteres encerrados entre apóstrofes)

Etiqueta (A/Z)

- Corresponden a caracteres alfabéticos que van de la A a la Z y su largo máximo es de 10 caracteres.

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC1 (Fecha 1 cuadrante A)



Campos de Salida de datos:

- T3FEC1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC1 (Fecha 1 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC1 (Fecha 1 cuadrante B)



Campos de Salida de datos:

- T3FEC1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC1 (Fecha 1 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC1 (Fecha 1 cuadrante C)



Campos de Salida de datos:

- T3FEC1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC1 (Fecha 1 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC2 (Fecha 2 cuadrante A)



Campos de Salida de datos:

- T3FEC2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC2 (Fecha 2 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC2 (Fecha 2 cuadrante B)



Campos de Salida de datos:

- T3FEC2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC2 (Fecha 2 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC2 (Fecha 2 cuadrante C)



Campos de Salida de datos:

- T3FEC2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC2 (Fecha 2 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC3 (Fecha 3 cuadrante A)



Campos de Salida de datos:

- T3FEC3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC3 (Fecha 3 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC3 (Fecha 3 cuadrante B)



Campos de Salida de datos:

- T3FEC3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC3 (Fecha 3 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *FEC3 (Fecha 3 cuadrante C)



Campos de Salida de datos:

- T3FEC3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0FEC3 (Fecha 3 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT1 (Comprobante 1 cuadrante A)



Campos de Salida de datos:

- T3CPT1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0NCPT (Numero de Comprobante del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT1 (Comprobante 1 cuadrante B)



Campos de Salida de datos:

- T3CPT1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0NCPT (Numero de Comprobante del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT1 (Comprobante 1 cuadrante C)



Campos de Salida de datos:

- T3CPT1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0NCPT (Numero de Comprobante del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT2 (Comprobante 2 cuadrante A)



Campos de Salida de datos:

- T3CPT2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- XONREF (Numero de Referencia del cuadrante A)
- XOSREF (Sub-numero de referencia del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT2 (Comprobante 2 cuadrante B)



Campos de Salida de datos:

- T3CPT2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- XONREF (Numero de Referencia del cuadrante B)
- XOSREF (Sub-numero de Referencia del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT2 (Comprobante 2 cuadrante C)



Campos de Salida de datos:

- T3CPT2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- XONREF (Numero de Referencia del cuadrante C)
- XOSREF (Sub-numero de Referencia del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT3 (Comprobante 3 cuadrante A)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0DREF (Dato Complementario del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT3 (Comprobante 3 cuadrante B)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0DREF (Dato Complementario del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CPT3 (Comprobante 3 cuadrante C)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0DREF (Dato Complementario del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS1 (Tasa 1 cuadrante A)



Campos de Salida de datos:

- T3TAS1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS1 (Tasa 1 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS1 (Tasa 1 cuadrante B)



Campos de Salida de datos:

- T3TAS1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS1 (Tasa 1 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS1 (Tasa 1 cuadrante C)



Campos de Salida de datos:

- T3TAS1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS1 (Tasa 1 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS2 (Tasa 2 cuadrante A)



Campos de Salida de datos:

- T3TAS2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS2 (Tasa 2 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS2 (Tasa 2 cuadrante B)



Campos de Salida de datos:

- T3TAS2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS2 (Tasa 2 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS2 (Tasa 2 cuadrante C)



Campos de Salida de datos:

- T3TAS2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS2 (Tasa 2 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS3 (Tasa 3 cuadrante A)



Campos de Salida de datos:

- T3TAS3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS3 (Tasa 3 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS3 (Tasa 3 cuadrante B)



Campos de Salida de datos:

- T3TAS3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS3 (Tasa 3 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *TAS3 (Tasa 3 cuadrante C)



Campos de Salida de datos:

- T3TAS3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0TAS3 (Tasa 3 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP1 (Importe 1 cuadrante A)



Campos de Salida de datos:

- T3IMP1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP1 (Importe 1 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP1 (Importe 1 cuadrante B)



Campos de Salida de datos:

- T3IMP1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP1 (Importe 1 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP1 (Importe 1 cuadrante C)



Campos de Salida de datos:

- T3IMP1 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP1 (Importe 1 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP2 (Importe 2 cuadrante A)



Campos de Salida de datos:

- T3IMP2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP2 (Importe 2 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP2 (Importe 2 cuadrante B)



Campos de Salida de datos:

- T3IMP2 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP2 (Importe 2 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP3 (Importe 3 cuadrante A)



Campos de Salida de datos:

- T3IMP3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP3 (Importe 3 del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP3 (Importe 3 cuadrante B)



Campos de Salida de datos:

- T3IMP3 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP3 (Importe 3 del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IMP3 (Importe 3 cuadrante C)



Campos de Salida de datos:

- T3IMP3 (Descripción del campo para la transacción)

Campos de Entrada/Salida de datos:

- X0IMP3 (Importe 3 del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CAN (Cantidad cuadrante A)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

Campos de Entrada/Salida de datos:

- X0CANT (Cantidad del cuadrante A)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CAN (Cantidad cuadrante B)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

Campos de Entrada/Salida de datos:

- XOCANT (Cantidad del cuadrante B)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *CAN (Cantidad cuadrante C)



Campos de Salida de datos:

- T3CPT3 (Descripción del campo para la transacción)

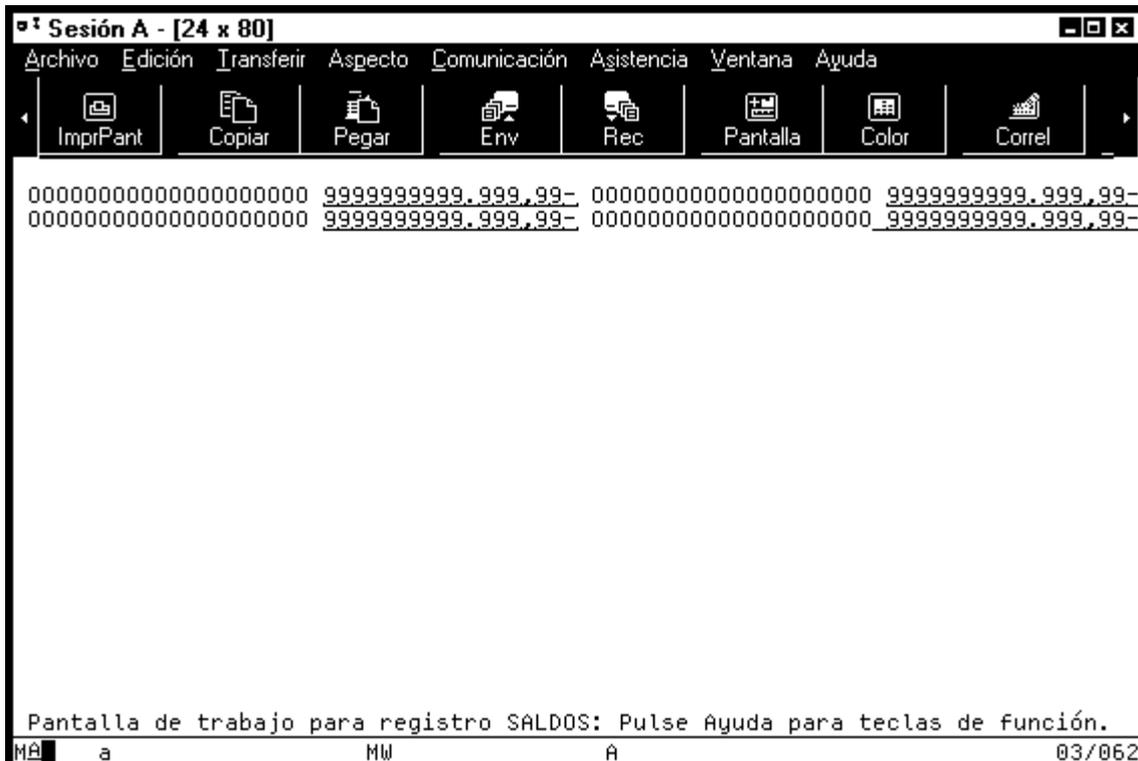
Campos de Entrada/Salida de datos:

- XOCANT (Cantidad del cuadrante C)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *SDO (Saldos)



Campos de Salida de datos:

- TOSDO1 (Descripción del campo para la transacción)
- TOSDO2 (Descripción del campo para la transacción)
- TOSDO3 (Descripción del campo para la transacción)
- TOSDO4 (Descripción del campo para la transacción)

Campos de Entrada/Salida de datos:

- COSDO1 (Saldo 1 – Ej. Saldo Contable)
- COSDO2 (Saldo 2 – Ej. Saldo Operativo)
- COSDO3 (Saldo 3 – Ej. Acuerdos)
- COSDO4 (Saldo 4 – Ej. Autorizaciones)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *PLF2 (Determina fecha de vencimiento)

Si se ingresa días calcula la fecha si no considera la tipeada en fecha de vencimiento

Sesión A - [24 x 80]

Archivo Edición Transferir Aspecto Comunicación Asistencia Ventana Ayuda

ImprPant Copiar Pegar Env Rec Pantalla Color Correl

Importe del Capital: 9999999.999.999,99- Cantidad de Dias.....: 99999
Fecha de Vencimiento.: 99/99/9999

Pantalla de trabajo para registro PLFJ02: Pulse Ayuda para teclas de función.

MA a MW A 03/078

Campos de Entrada/Salida de datos:

- X0IMP1 (Importe 1 – Capital)
- X0CNT\$ (Cantidad – Cantidad de días)
- X0FEC2 (Fecha – Fecha de Vencimiento)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *PLF3 (tasa, representante con documento)



Campos de Salida de datos:

- X0IMP1 (Importe 1 – Capital)
- X0CNT\$ (Cantidad – Cantidad de días)
- X0FEC2 (Fecha – Fecha de Vencimiento)
- T3TAS1 (Descripción del campo tasa para la transacción)

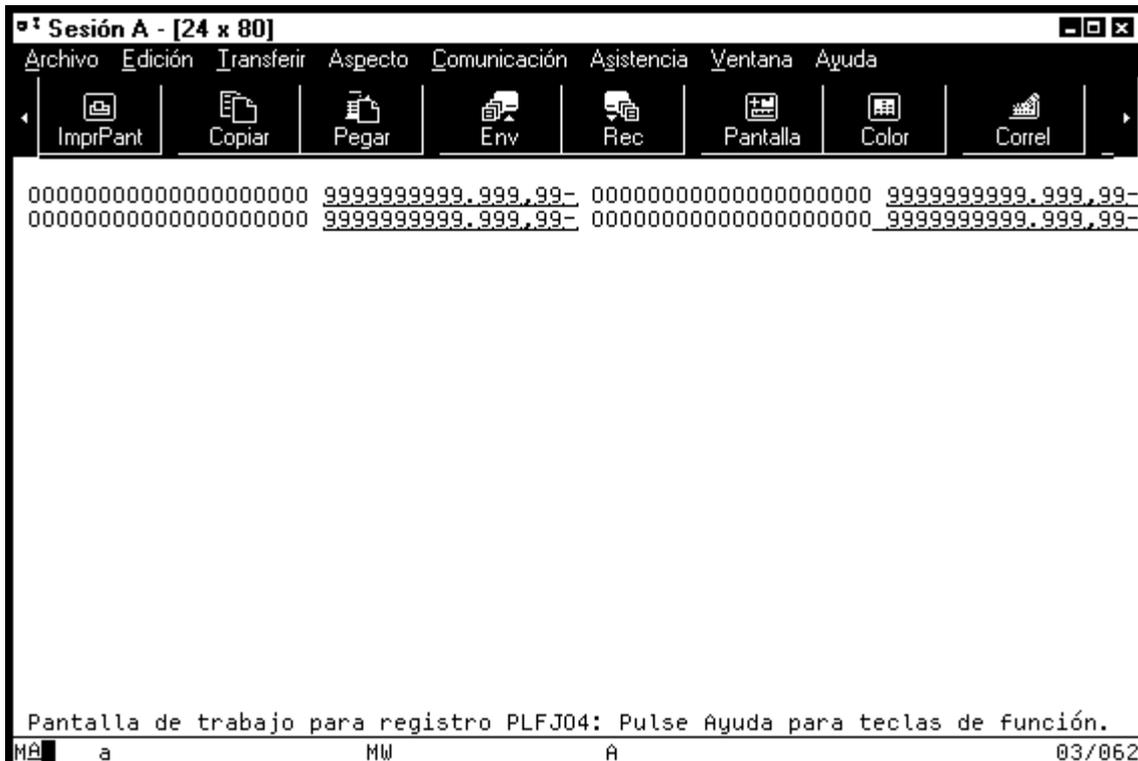
Campos de Entrada/salida de datos:

- X0TAS1 (Tasa Nominal Anual)
- X2DENO (Representante)
- X2TDOC (Tipo de Documento)
- X2NDOC (Numero de Documento)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *PLF4 (Interés, Retenciones y neto)



Campos de Salida de datos:

- T3IMP2 (Descripción del campo para la transacción)
- T3IMP3 (Descripción del campo para la transacción)
- T3IMP4 (Descripción del campo para la transacción)
- T3IMP5 (Descripción del campo para la transacción)

Campos de Entrada/salida de datos:

- X0IMP2 (Importe 2 - Interés)
- X0IMP3 (Importe 3 - Retención 1)
- X0IMP4 (Importe 4 - Retención 2)
- X0IMP5 (Importe 5 - Neto)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

Formato = *IDCLI (Identificación del cliente)

Doc. Identidad: BB_999999999999 Nombre: 00000000000000000000000000000000 Sexo:

Domici: 00000000000000000000000000000000 Puerta: 66666 Manzana: 00 Block: 00
Barrio: 00000000000000000000000000000000 Piso: 00 Departamento: 000 Casa: 000
Locali: 00000000000000000000000000000000 Cd.Pos: 66666 Provincia: 00 Zona:_00000

Pantalla de trabajo para registro IDCLIE: Pulse Ayuda para teclas de función.

MA a MW A 08/075

Campos de Entrada/salida de datos:

- X2TDOC (Tipo de Documento)
- X2NDOC (Numero de Documento)
- X3DENO (Nombre)
- X3DOMI (Domicilio)
- X3NPTA (Puerta)
- X3MZNA (Manzana)
- X3BLCK (Block)
- X3BRIO (Barrio)
- X3PISO (Piso)
- X3DEPT (Departamento)
- X3CASA (Casa)
- X3LOCA (Localidad)
- X3PROV (Provincia)

APLICATIVO BANCARIO GENERICO

Sentencias de Codificación

- X3ZONA (Zona)

Formato = Invocado por la función PRTFOR para impresión de comprob. con numeración.

En caso de ser prenumerados es que solicita, mediante este formato, la confirmación del número electrónico con el físico.

