# inicio

## ! Definicion de las bases de datos externas

ArcExp file, pre(TExp), driver('TOPSPEED'), create, reclaim, name('Est\_VtaD')

LLP\_Id key(TipoReg, TipoNotaCred, Cod\_Almacen, Cod\_Vendedor, Cod\_Facturista, Anios, Meses, Cod\_Cliente, Cod\_Articulo)

LLS\_MA key(TExp:Meses,TExp:Anios), dup

LLS\_MAAA Key(TExp:Meses,TExp:Anios,TExp:Cod\_Articulo,TExp:Cod\_Almacen), dup

LLS\_ArtAlmDate Key(TExp:Cod\_Articulo,TExp:Cod\_Almacen,TExp:Meses,TExp:Anios), dup

LLS\_CA key(TExp:Cod\_Cliente,TExp:Cod\_Articulo), dup

Record Record

TipoReg string(18)

Cod\_Almacen like(ALM:NAlm)

Almacen like(ALM:Descripcion)

Cod\_Articulo like(DFAC:CArt)

Unidades like(DFAC:Cantidad)

Costo\_Venta decimal(11, 2)

Costo\_Reposic decimal(11, 2)

Descuento decimal(11, 2)

Costo\_NetoVta decimal(11, 2)

Venta\_Sin\_IVA decimal(11, 2) !! Esta es la que hay que tomar para el calculo de comisiones fumado

IVA decimal(11, 2)

Venta\_Con\_IVA decimal(11, 2)

Utilidad decimal(11, 2)

DesctosNCredito decimal(11, 2) ! No Tamar en

Meses string(2)

Anios string(4)

Cod\_Cliente like(FAC:NCte)

Cod\_Vendedor like(FAC:NVen)

Cod\_Facturista like(FAC:NVen\_Fac)

TipoNotaCred string(17)

Cliente like(CTE:Nombre)

Ciudad like(CTE:Ciudad)

ZonaCobranza like(CTE:ZonCobran)

ClasifCliente like(CTE:CClasif)

Vendedor like(VEN:Nombre)

Facturista like(VEN:Nombre)

Articulo like(ART:DescNormal)

Cod\_Grupo like(ART:CGpo)

Cod\_Linea like(ART:Linea)

UnidadVenta like(ART:UniVenta)

ClasifArticulo like(ART:Clasificacion)

Cod\_Proveedor like(ART:NPrv)

Linea like(CLI:Descripcion)

Grupo like(GPO:Descripcion)

Proveedor like(PRV:Nombre)

Clasif\_Hits like(Pesa:Clasif\_Hits)

Clasif\_Vtas like(Pesa:Clasificacion)

end

end

ArcVtas file, pre(TVta), driver('TOPSPEED'), create, reclaim, name('Com\_Vtas')

LLP\_Id key(TVta:Nven,TVta:NCte, TVta:FcFecha), Dup

LLS\_Id key(TVta:Nven,TVta:TipoCte,TVta:NCte, TVta:FcFecha), Dup

Record Record

NCte like(FAC:NCte)

NVen like(Fac:NVen)

LFolio like(ADXC:LFolio)

NFolio like(ADXC:NFolio)

FcFecha Long

ImpSinIva Decimal(12,2)

Comision Decimal(12,2)

Antiguedad Decimal(12,2)

ImpComision Decimal(12,2)

Dias\_Atraso Decimal(12,2)

TipoCte String(3)

end

end

ArcVtasNtas file, pre(TVN), driver('TOPSPEED'), create, reclaim, name('Vtas\_Net')

LLP\_Id key(TVN:Nven,TVN:Tipo, TVN:Nivel, TVN:Valor)

Record Record

NVen Like(Fac:NVen)

Tipo Like(TCV:Tipo)

Nivel Like(TCV:Nivel)

Valor Like(TCV:Valor)

Venta Decimal(12,2)

end

end

ArcVtasNtasC file, pre(TVNC), driver('TOPSPEED'), create, reclaim, name('Vtas\_NetC')

LLP\_Id key(TVNC:Ciudad,TVNC:Tipo, TVNC:Nivel, TVNC:Valor)

Record Record

Ciudad Like(ALM:Ciudad)

Tipo Like(TCV:Tipo)

Nivel Like(TCV:Nivel)

Valor Like(TCV:Valor)

Venta Decimal(12,2)

end

end

!!! Tabla Temporal de Folios de Pagos

Pagos file, pre(TPA), driver('TOPSPEED'), create, reclaim, name('Pagos\_Com.Tps')

LLP\_Id key(TPA:LFolioF,TPA:NFolioF) dup

LLP\_NVen key(TPA:NVen) dup

Record Record

LFolioF Like(ADXC:LFolio)

NFolioF Like(ADXC:NFolio)

LFolioP Like(ADXC:LFolio)

NFolioP Like(ADXC:NFolio)

FcFecha Long

NVen Like(Fac:NVen)

Importe Like(ADXC:Importe)

PrecNeto Like(ADXC:Importe)

CostNeto Like(ADXC:Importe)

Utilidad Like(ADXC:Importe)

end

end

! Tabla Temporal de Cotizaciones por vendedores CAT

CotVen FILE, pre(CV1), DRIVER('TOPSPEED'), create, NAME('CotVen.TPS')

CV1:LLP\_VEN\_FAC KEY(+CV1:NVEN\_FAC),NOCASE,OPT,PRIMARY

CV1:LLS\_CIUDAD\_NVEN KEY(+CV1:CIUDAD,+CV1:NVEN\_FAC),DUP,NOCASE,OPT

CV1:RECORD RECORD

CV1:NVEN\_FAC LONG

CV1:CIUDAD STRING(25)

CV1:PARTIDAS LONG

END

END

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# ! Inicio del proceso

!! Abrir Tablas

DO SR\_AbrirTablas

If Loc:Error = True then Cycle.

!! Comisiones por Vendedor

Close(Pagos)

Create(Pagos)

if errorcode() then

message('No se Puede Crear el Archivo de Pagos por Vendedor. |Error: ' & error())

cycle

end

open(Pagos, Readwrite+denyall)

if errorcode() then

message('No se Puede Abrir el Archivo de Pagos por Vendedor. |Error: ' & error())

cycle

end

Do SR\_abrirT\_Most

If Loc:Error = True Then Cycle.

! Abrir Temporal Vendedores CAT

Do SR\_abrir\_CotVen

If Loc:Error = True then Cycle.

!Iniciar Transaccion

logout(1, ArcVtas, ArcVtasNtas, ArcVtasNtasC, pagos, ELRTCV, ELRTCC, ElVen, ELMVM, ELMVMC, COTVEN)

If ErrorCode() then

message('No se logró iniciar la transacción, favor de intertarlo mas tarde')

cycle

End

## ! Fase 1 Comisiones

Do Sr\_ComVtas

## ! Fase 2 Ventas

Do Sr\_Ventas\_Ven

## ! Fase 3 Rangos para Resultados de venta

Do Sr\_Ran\_Com

## !! Comision segun Pagos

Do SR\_Generar\_Pag

## !! Metodologia para Vendedores de Mostrador

Do SR\_ComMostrador

## ! Metodología para Vendedores CAT

Do Sr\_CotVen

commit

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!Abrir Tablas

# SR\_AbrirTablas Routine

Loc:Error = False

!! DWH de Ventas

Close(ArcExp)

! CREATE(ArcExp)

! if errorcode()

! message('No se Puede Crear el Archivo de DHW. |' & |

! 'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

! Loc:Error = True

! exit

! end

open(ArcExp, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de DHW. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!! Agrupado de Comisiones por Vendedor

Close(ArcVtasNtas)

Create(ArcVtasNtas)

if errorcode()

message('No se Puede Crear el Archivo de Ventas Netas por Vendedor. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

open(ArcVtasNtas, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de Ventas Netas por Vendedor. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!! Agrupado de Comisiones por ciudad

Close(ArcVtasNtasC)

Create(ArcVtasNtasC)

if errorcode()

message('No se Puede Crear el Archivo de Ventas Netas por Ciudad. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

open(ArcVtasNtasC, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de Ventas Netas por Ciudad. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!! Archivo de Ventas

Close(ArcVtas)

create(ArcVtas)

if errorcode()

message('No se Puede Crear el Archivo de Exportación. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

open(ArcVtas, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de Exportación de Comisiones. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!! Eliminar la Tabla de Resultados Anteriores

!! Recorrer Ventas Registradas

!! Verificar el Resultado en las tablas si esta Registrado para eso

!! Guardar el REsultado en la Tabla de resultados

Close(ELRTCV)

Remove(ELRTCV)

!! Agrupado de Comisiones por Vendedor

Create(ELRTCV)

if errorcode()

message('No se Puede Crear el Archivo de Resultado de Comisiones. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

open(ELRTCV, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de Resultado de Comisiones. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

Close(ELRTCC)

Remove(ELRTCC)

!! Agrupado de Comisiones por Vendedor

Create(ELRTCC)

if errorcode()

message('No se Puede Crear el Archivo de Resultado de Comisiones por Ciudad. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

open(ELRTCC, Readwrite+denyall)

if errorcode()

message('No se Puede Abrir el Archivo de Resultado de Comisiones por Ciudad. |' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Rutina para abrir las tablas para este Proceso

# SR\_abrirT\_Most Routine

Loc:Error = False

!! Crear Tabla Externa

Close(ELMVM)

create(ELMVM)

if errorcode()

message('No se Pudo Crear el Archivo de Salida(Capturistas).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Close(ELMVM)

Loc:Error = True

exit

end

!! Abrir Tabla creada

open(ELMVM, Readwrite+denyall)

if errorcode()

message('No se Pudo Abrir el Archivo de Salida(Capturistas).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!! Crear Tabla Externa

Close(ELMVMC)

create(ELMVMC)

if errorcode()

message('No se Pudo Crear el Archivo de Salida(Ciudades).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Close(ELMVM)

Loc:Error = True

exit

end

!! Abrir Tabla creada

open(ELMVMC, Readwrite+denyall)

if errorcode()

message('No se Pudo Abrir el Archivo de Salida(Ciudades).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Close(ELMVM)

Loc:Error = True

exit

end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Rutina para abrir las tablas para este Proceso

# SR\_abrir\_CotVen Routine

Loc:Error = False

!! Crear Tabla Externa

Close(COTVEN)

create(COTVEN)

if errorcode()

message('No se Pudo Crear el Archivo de Salida(Vendedores CAT).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Close(COTVEN)

Loc:Error = True

exit

end

!! Abrir Tabla creada

open(COTVEN, Readwrite+denyall)

if errorcode()

message('No se Pudo Abrir el Archivo de Salida(Vendedores CAT).|' & |

'Error: ' & error(), 'Error de Proceso', icon:hand, button:ok)

Loc:Error = True

exit

end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Rutina de comisiones por venta

# Sr\_ComVtas Routine

Data

Lof:Fecha Long

Code

Lof:Fecha= Date(CTRL:Cie\_MesActual,1,CTRL:Cie\_AnoActual)

Loc:RegCom = 0

Loc:RegCom2 = 0

Loc:RegCom3 = 0

Loc:RegCom4 = 0

Loc:RecAct2 = 0

Loc:RecAct = 0

Stream(ArcVtas)

!! Recorremos los Pagos

Clear(ADXC:Record)

ADXC:FcRegistro = Lof:Fecha

SET(ADXC:LLS\_FcRegistro\_NCte,ADXC:LLS\_FcRegistro\_NCte)

Loop

Next(ELADXC)

If ErrorCode() Then Break.

!! Tanteo

If ~((CLIP(ADXC:Cdoc) = 'P' AND ADXC:CConcepto = 15) OR (CLIP(ADXC:Cdoc) = 'SFC' AND ADXC:CConcepto = 19) OR |

(CLIP(ADXC:Cdoc) = 'P' AND ADXC:CConcepto = 6) OR (CLIP(ADXC:Cdoc) = 'SFC' AND ADXC:CConcepto = 15)) Then Cycle.

If Month(ADXC:FcRegistro) <> CTRL:Cie\_MesActual Or |

year(ADXC:FcRegistro) <> CTRL:Cie\_AnoActual Then Break.

! Busco si el folio de la factura contiene un anticipo

!1. band. en: no imprimir detalle

!2.. inic. totales

Loc:bnd = false

Loc:Importe = 0

Loc:PrcComision= 0

Loc:ImpComision= 0

clear(Fac:Record)

FAC:LFolio = ADXC:ALFolio

FAC:NFolio = ADXC:ANFolio

get(Elfac, FAC:LLP\_LFolio\_NFolio)

if errorcode() then Cycle .

If FAC:Estado = 0 Then Cycle .!! No Tomar Facturas Canceladas

!! Validación para Capturista

IF (CLIP(ADXC:Cdoc) = 'P' AND ADXC:CConcepto = 6) THEN

!! Verificamos Capturista

CLEAR(VEN:Record)

VEN:NVen = FAC:NVen\_Fac

GET(ELVEN, VEN:LLP\_NVendedor)

IF ERRORCODE() THEN CYCLE .

!! Verificar que sea vendedor, sino, omitir

IF VEN:Tipo\_Vend <> 1 THEN CYCLE .

END

!! Buscamos el Vendedor

CLEAR(VEN:Record)

VEN:NVen = FAC:NVen

GET(ELVEN,VEN:LLP\_NVendedor)

If ErrorCode() Then Clear(Ven:Record).

!! Buscamos el Almacen Asignado al Vendedor

CLEAR(Alm:Record)

ALM:NAlm = VEN:NAlm

GET(ELAlm,ALM:LLP\_NAlmacen)

If ErrorCode() Then Clear(Alm:Record).

!! Buscamos el Cliente

!! Checamos si tiene la bandera de Considerar en Comisiones

CLEAR(CTE:Record)

CTE:NCte = ADXC:NCte

GET(ElCte,CTE:LLP\_NCliente)

If ErrorCode() Then Clear(Cte:Record).

If ~CTE:Bnd\_ConCom then Cycle .

if FAC:NFolio then

If ~FAC:Bnd\_AntFac !! Si es un anticipo no se toma en cuenta

clear(DFac:Record)

DFAC:Consecutivo= FAC:Consecutivo

DFAC:CArt = 'V-ANTICIPO'

get(ElDfac, DFAC:LLS\_Consecutivo\_CArt)

if errorcode() then

Loc:bnd=true

!! calcular los importes

Loc:Importe = ADXC:ImpAplicar / (1 + (CTRL:Grl\_IVA / 100))

!! calcular el total de la comision

If CTE:Prc\_Comision > 0 !! Si se capturo en la clientes

Loc:PrcComision = CTE:Prc\_Comision

Else

if (today() - CTE:FcAlta) <= VEN:NDiasAnt

Loc:PrcComision = VEN:Prc\_Com1

else

Loc:PrcComision = VEN:Prc\_Com2

end

End

!! Buscamos CXC

Clear(CXC:Record)

CXC:LFolio = FAC:LFolio

CXC:NFolio = FAC:NFolio

GET(ELCXC, CXC:LLP\_LFolio\_NFolio)

If ErrorCode() Then Clear(CXC:Record).

Loc:ImpComision = Loc:Importe \* (Loc:PrcComision/100)

Loc:PrcAntig = 0

Loc:Dias = ADXC:FcDeposito - CXC:FcVenc !(FAC:FcFactura+FAC:DiasPlazo) !Today()- (FAC:FcFactura+FAC:DiasPlazo)

Clear(TPCV:Record)

TPCV:NVen = VEN:NVen

SET(TPCV:LLP\_NVen\_DiasPzo,TPCV:LLP\_NVen\_DiasPzo)

Loop

Next(ELTPCV)

If ErrorCode() Then Break.

If TPCV:NVen<>VEN:NVen Then Break.

!! Reducir comision segun dias transcurridos

If Loc:Dias <= TPCV:DiasPzo

Loc:ImpComision = Loc:ImpComision \*(1-(TPCV:PrComision/100))

Loc:PrcAntig = TPCV:PrComision

Break

End

End

Loc:TgImporte += Loc:Importe

Loc:TgComision += Loc:ImpComision

end

End

end

Clear(TVta:Record)

TVta:NCte = FAC:NCte

TVta:NVen = Fac:NVen

TVta:LFolio = ADXC:LFolio

TVta:NFolio = ADXC:NFolio

TVta:FcFecha = ADXC:FcDeposito

TVta:ImpSinIva = Loc:Importe

TVta:Comision = Loc:PrcComision

TVta:Antiguedad = Loc:PrcAntig

TVta:ImpComision = Loc:ImpComision

TVta:Dias\_Atraso = Loc:Dias

!! Clasificacion de Cliente Nuevo, Normal, Bajo Margen

if CTE:Bnd\_CteNvo !!CTE:FcAlta+ VEN:NDiasAnt > Today()

TVta:TipoCte = 'NVO'

Elsif CTE:Prc\_Comision <= 0

TVta:TipoCte = 'NOR'

Elsif CTE:Prc\_Comision > 0

TVta:TipoCte = 'BMG'

End

ADD(ArcVtas)

If ~ErrorCode()

Loc:RegCom+=1

Display(?Loc:RegCom)

End

End

Flush(ArcVtas)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! 1.- Recorrer DWH de Ventas recopilando las ventas

# Sr\_Ventas\_Ven Routine

Data

Lof:Fecha Long

Code

Lof:Fecha= Date(CTRL:Cie\_MesActual,1,CTRL:Cie\_AnoActual)

Stream(ArcVtasNtas)

Stream(ArcVtasNtasC)

Clear(TExp:Record)

TExp:Meses = Format(CTRL:Cie\_MesActual,@n02)

TExp:Anios = CTRL:Cie\_AnoActual

SET(TExp:LLS\_MA,TExp:LLS\_MA)

Loop

!1.- Sig. Registro

!2.- Caso de Error

!3.- Rango de Meses

!4.- Agrupacion segun cada tipo

Next(ArcExp)

If ErrorCode() Then Break.

If TExp:Meses<>Format(CTRL:Cie\_MesActual,@n02) or |

TExp:Anios<>CTRL:Cie\_AnoActual Then Break.

!! Buscamos Cliente

CLEAR(CTE:Record)

CTE:NCte = TExp:Cod\_Cliente

GET(ElCte,CTE:LLP\_NCliente)

If ErrorCode() Then Cycle.

!! validacion de Solo clientes normales

!if CTE:Prc\_Comision>0 Then Cycle.

!! Buscamos el Vendedor

CLEAR(VEN:Record)

VEN:NVen = TExp:Cod\_Vendedor

GET(ELVEN,VEN:LLP\_NVendedor)

If ErrorCode() Then Cycle.

!! Buscamos el Almacen Asignado al Vendedor

CLEAR(ALM:Record)

ALM:NAlm = VEN:NAlm

GET(ELAlm,ALM:LLP\_NAlmacen)

If ErrorCode() Then Clear(Alm:Record).

!! Buscamos el Articulo

CLEAR(ART:Record)

ART:CArt= TExp:Cod\_Articulo

GET(ELArt,ART:LLP\_CArticulo)

If ErrorCode() Then Clear(Art:Record).

If TExp:TipoReg <> 'OTRAS N.A.'

! Tipo Ventas

Do Sr\_Ventas

! Tipo Ventas Especificas

Do Sr\_Vtas\_Esp

! Tipo Utilidad

Do Sr\_Utilidad

End

Loc:RegCom2+=1

Display(?Loc:RegCom2)

End

Flush(ArcVtasNtas)

Flush(ArcVtasNtasC)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! 1 Agrupamiento General por Vendedor y Ciudad

# Sr\_Ventas Routine

!! Agrupamiento por Vendedor

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'A'

TVN:Nivel = 'V'

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'A'

TVN:Nivel = 'V'

TVN:Valor = ''

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

!! Agrupamiento por Vendedor y ciudad

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'D'

TVN:Nivel = 'V'

TVN:Valor = ALM:Ciudad

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'D'

TVN:Nivel = 'V'

TVN:Valor = ALM:Ciudad

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

!! Agrupamiento por ciudad Solo Vendedores de Mostrador

If Instring('MOSTRADOR',VEN:Nombre,1,1)<>0

Clear(TVNC:Record)

TVNC:Ciudad = ALM:Ciudad

TVNC:Tipo = 'A'

TVNC:Nivel = 'C'

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad = ALM:Ciudad

TVNC:Tipo = 'A'

TVNC:Nivel = 'C'

TVNC:Valor = ''

TVNC:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtasC)

End

!! Agrupamiento por Capturista

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Facturista

TVN:Tipo = 'A'

TVN:Nivel = 'F'

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Facturista

TVN:Tipo = 'A'

TVN:Nivel = 'F'

TVN:Valor = ''

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

End

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! 2 Ventas Especificas Vendedor y Ciudad

# Sr\_Vtas\_Esp Routine

! Proveedor

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'P'

TVN:Valor = TExp:Cod\_Proveedor

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'P'

TVN:Valor = TExp:Cod\_Proveedor

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

! Marca

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'M'

TVN:Valor = ART:CMar

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'M'

TVN:Valor = ART:CMar

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

! Linea

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'L'

TVN:Valor = ART:Linea

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'L'

TVN:Valor = ART:Linea

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

! Articulo

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'A'

TVN:Valor = TExp:Cod\_Articulo

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'B'

TVN:Nivel = 'A'

TVN:Valor = TExp:Cod\_Articulo

TVN:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtas)

End

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

!!! Ventas por Ciudad

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

! Proveedor

Clear(TVNC:Record)

TVNC:Ciudad = ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'P'

TVNC:Valor = TExp:Cod\_Proveedor

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'P'

TVNC:Valor = TExp:Cod\_Proveedor

TVNC:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtasC)

End

! Marca

Clear(TVNC:Record)

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'M'

TVNC:Valor = ART:CMar

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'M'

TVNC:Valor = ART:CMar

TVNC:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtasC)

End

! Linea

Clear(TVNC:Record)

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'L'

TVNC:Valor = ART:Linea

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'L'

TVNC:Valor = ART:Linea

TVNC:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtasC)

End

! Articulo

Clear(TVNC:Record)

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'A'

TVNC:Valor = TExp:Cod\_Articulo

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad= ALM:Ciudad

TVNC:Tipo = 'B'

TVNC:Nivel = 'A'

TVNC:Valor = TExp:Cod\_Articulo

TVNC:Venta = TExp:Venta\_Sin\_IVA

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Venta\_Sin\_IVA

PUT(ArcVtasNtasC)

End

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Utilidad

# Sr\_Utilidad Routine

Clear(TVN:Record)

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'C'

TVN:Nivel = 'V'

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode()

TVN:NVen = TExp:Cod\_Vendedor

TVN:Tipo = 'C'

TVN:Nivel = 'V'

TVN:Venta = TExp:Utilidad

ADD(ArcVtasNtas)

Else

TVN:Venta += TExp:Utilidad

PUT(ArcVtasNtas)

End

If Instring('MOSTRADOR',VEN:Nombre,1,1)<>0

!! Utilidad por ciudad

Clear(TVNC:Record)

TVNC:Ciudad = ALM:Ciudad

TVNC:Tipo = 'C'

TVNC:Nivel = 'C'

GET(ArcVtasNtasC,TVNC:LLP\_Id)

If ErrorCode()

TVNC:Ciudad = ALM:Ciudad

TVNC:Tipo = 'C'

TVNC:Nivel = 'C'

TVNC:Valor = ''

TVNC:Venta = TExp:Utilidad

ADD(ArcVtasNtasC)

Else

TVNC:Venta += TExp:Utilidad

PUT(ArcVtasNtasC)

End

End

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Fase 3 Dar porcentajes de Comision segun las tablas Registradas \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Sr\_Ran\_Com Routine

Data

Lof:LimInf Decimal(12,2)

Lof:Promesa Decimal(12,2)

Lof:Bnd\_Enc Byte

Code

Stream(ELRTCV)

Clear(TCVC:Record)

SET(TCVC:LLP\_NVen\_Tipo\_Nivel\_Val,TCVC:LLP\_NVen\_Tipo\_Nivel\_Val)

Loop

!1 Recorremos las tablas Registradas

!2 Solo las que tengan Venta

!3 Buscamos si hay algun valor registrado en ventas

!4 Recorremos los Detalles de la Tabla para conocer el porcentaje que le toca

Next(ELTCVC)

If ErrorCode() Then Break.

If TCVC:Tipo <> 'C' !! Diferente a tabla de utilidad

!! buscamos resultado actual de la tabla

TVN:NVen = TCVC:NVen

TVN:Tipo = TCVC:Tipo

TVN:Nivel= TCVC:Nivel

TVN:Valor= TCVC:Valor

GET(ArcVtasNtas, TVN:LLP\_Id)

If ErrorCode() Then Cycle.

!! Tomamos el valor segun el mes para sacar el porcentaje cumplido

Execute CTRL:Cie\_MesActual

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Enero)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Febrero)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Marzo)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Abril)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Mayo)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Junio)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Julio)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Agosto)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Septiembre)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Octubre)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Noviembre)\*100

Loc:PrcVta = (TVN:Venta/ TCVC:Obj\_Diciembre)\*100

End

Else

!! Buscamos la Ventan neta del Vendedor

TVN:NVen = TCVC:NVen

TVN:Tipo = 'A'

TVN:Nivel= 'V'

TVN:Valor= ''

GET(ArcVtasNtas, TVN:LLP\_Id)

If ErrorCode() Then Cycle.

!! guadamos la venta de utilidad en variable porque modificaremos el puntero

Loc:PrcVta = TVN:Venta

!! buscamos resultado actual de la tabla

TVN:NVen = TCVC:NVen

TVN:Tipo = TCVC:Tipo

TVN:Nivel= TCVC:Nivel

TVN:Valor= TCVC:Valor

GET(ArcVtasNtas, TVN:LLP\_Id)

If ErrorCode() Then Cycle.

Loc:PrcVta = (Loc:PrcVta/TVN:Venta)\*100

!Loc:PrcVta = (TVN:Venta/Loc:PrcVta)\*100 antes antes

End

!! Si el Tipo es igual a Ventas

If TCVC:Tipo = 'A'

Lof:Bnd\_Enc = True

!! Guardamos Objetivo en limite inferior por si es tabla de ventas

Execute CTRL:Cie\_MesActual

Lof:LimInf = TCVC:Obj\_Enero

Lof:LimInf = TCVC:Obj\_Febrero

Lof:LimInf = TCVC:Obj\_Marzo

Lof:LimInf = TCVC:Obj\_Abril

Lof:LimInf = TCVC:Obj\_Mayo

Lof:LimInf = TCVC:Obj\_Junio

Lof:LimInf = TCVC:Obj\_Julio

Lof:LimInf = TCVC:Obj\_Agosto

Lof:LimInf = TCVC:Obj\_Septiembre

Lof:LimInf = TCVC:Obj\_Octubre

Lof:LimInf = TCVC:Obj\_Noviembre

Lof:LimInf = TCVC:Obj\_Diciembre

End

Lof:Promesa = Loc:PrcVta

Else

Lof:Bnd\_Enc = False

Lof:LimInf = 0

Clear(TCV:Record)

TCV:NVen = TVN:NVen

TCV:Tipo = TVN:Tipo

TCV:Nivel = TVN:Nivel

TCV:Valor = TVN:Valor

TCV:Objetivo = 0 !TVN:Venta

SET(TCV:LLP\_NVen\_Tipo\_Nivel\_Val\_Obj,TCV:LLP\_NVen\_Tipo\_Nivel\_Val\_Obj)

Loop

Next(ELTCV)

If ErrorCode() Then Break.

If TCV:NVen <> TVN:NVen Or |

TCV:Tipo <> TVN:Tipo Or |

TCV:Nivel<> TVN:Nivel Or |

TCV:Valor<> TVN:Valor Then Break.

Lof:Bnd\_Enc = True

!! Guardamos Resultado de la Tabla

IF ~(Loc:PrcVta>=Lof:LimInf And Loc:PrcVta<TCV:Objetivo)

Lof:LimInf = TCV:Objetivo

Lof:Promesa= TCV:Promesa

Cycle

End

Break

End

!! Se Encontro un Limite en la tabla

End

IF Lof:Bnd\_Enc !and Lof:LimInf

Clear(RTCV:Record)

RTCV:NVen = TVN:NVen

RTCV:Tipo = TVN:Tipo

RTCV:Nivel = TVN:Nivel

RTCV:Valor = TVN:Valor

RTCV:Objetivo= Lof:LimInf

RTCV:Venta = TVN:Venta

RTCV:Promesa = Lof:Promesa

ADD(ELRTCV)

If ~ErrorCode()

Loc:RecAct+=1

Display(?Loc:RecAct)

End

End

Loc:RegCom3+=1

Display(?Loc:RegCom3)

End

Flush(ELRTCV)

Stream(ELRTCC)

Clear(TCCC:Record)

SET(TCCC:LLP\_Ciu\_Tipo\_Nivel\_Val,TCCC:LLP\_Ciu\_Tipo\_Nivel\_Val)

Loop

!1 Recorremos las tablas Registradas

!2 Solo las que tengan Venta

!3 Buscamos si hay algun valor registrado en ventas

!4 Recorremos los Detalles de la Tabla para conocer el porcentaje que le toca

Next(ELTCCC)

If ErrorCode() Then Break.

If TCCC:Tipo <> 'C' !! Diferente a tabla de utilidad

Clear(TVNC:Record)

TVNC:Ciudad = TCCC:Ciudad

TVNC:Tipo = TCCC:Tipo

TVNC:Nivel = TCCC:Nivel

TVNC:Valor = TCCC:Valor

GET(ArcVtasNtasC, TVNC:LLP\_Id)

If ErrorCode() Then Cycle.

Loc:PrcVta = 0

!! Tomamos el valor segun el mes para sacar el porcentaje cumplido

Execute CTRL:Cie\_MesActual

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Enero)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Febrero)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Marzo)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Abril)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Mayo)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Junio)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Julio)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Agosto)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Septiembre)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Octubre)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Noviembre)\*100

Loc:PrcVta = (TVNC:Venta/ TCCC:Obj\_Diciembre)\*100

End

Else

!! Buscamos la Ventan neta del Vendedor

TVNC:Ciudad = TCCC:Ciudad

TVNC:Tipo = 'A'

TVNC:Nivel= 'C'

TVNC:Valor= ''

GET(ArcVtasNtasC, TVNC:LLP\_Id)

If ErrorCode() Then Cycle.

!! guadamos la venta de utilidad en variable porque modificaremos el puntero

Loc:PrcVta = TVNC:Venta

!! buscamos resultado actual de la tabla

TVNC:Ciudad = TCCC:Ciudad

TVNC:Tipo = TCCC:Tipo

TVNC:Nivel = TCCC:Nivel

TVNC:Valor = TCCC:Valor

GET(ArcVtasNtasC, TVNC:LLP\_Id)

If ErrorCode() Then Cycle.

Loc:PrcVta = (Loc:PrcVta/TVN:Venta)\*100

End

!!! Si es de Venta

If TCCC:Tipo = 'A'

Lof:Bnd\_Enc = True

!! Guardamos Objetivo en limite inferior por si es tabla de ventas

Execute CTRL:Cie\_MesActual

Lof:LimInf = TCCC:Obj\_Enero

Lof:LimInf = TCCC:Obj\_Febrero

Lof:LimInf = TCCC:Obj\_Marzo

Lof:LimInf = TCCC:Obj\_Abril

Lof:LimInf = TCCC:Obj\_Mayo

Lof:LimInf = TCCC:Obj\_Junio

Lof:LimInf = TCCC:Obj\_Julio

Lof:LimInf = TCCC:Obj\_Agosto

Lof:LimInf = TCCC:Obj\_Septiembre

Lof:LimInf = TCCC:Obj\_Octubre

Lof:LimInf = TCCC:Obj\_Noviembre

Lof:LimInf = TCCC:Obj\_Diciembre

End

Lof:Promesa = Loc:PrcVta

Else

Lof:Bnd\_Enc = False

Lof:LimInf = 0

Clear(TCC:Record)

TCC:Ciudad = TVNC:Ciudad

TCC:Tipo = TVNC:Tipo

TCC:Nivel = TVNC:Nivel

TCC:Valor = TVNC:Valor

TCC:Objetivo = 0 !TVNC:Venta

SET(TCC:LLP\_Ciu\_Tipo\_Nivel\_Val\_Obj,TCC:LLP\_Ciu\_Tipo\_Nivel\_Val\_Obj)

Loop

Next(ELTCC)

If ErrorCode() Then Break.

If TCC:Ciudad<> TVNC:Ciudad Or |

TCC:Tipo <> TVNC:Tipo Or |

TCC:Nivel <> TVNC:Nivel Or |

TCC:Valor <> TVNC:Valor Then Break.

Lof:Bnd\_Enc = True

!! Guardamos Resultado de la Tabla

IF ~(Loc:PrcVta>=Lof:LimInf And Loc:PrcVta<TCC:Objetivo)

Lof:LimInf = TCC:Objetivo

Lof:Promesa= TCC:Promesa

Cycle

End

Break

End

End

!! Se Encontro un Limite en la tabla

IF Lof:Bnd\_Enc

Clear(RTCC:Record)

RTCC:Ciudad = TVNC:Ciudad

RTCC:Tipo = TVNC:Tipo

RTCC:Nivel = TVNC:Nivel

RTCC:Valor = TVNC:Valor

RTCC:Objetivo= Lof:LimInf

RTCC:Venta = TVNC:Venta

RTCC:Promesa = Lof:Promesa

ADD(ELRTCC)

If ~ErrorCode()

Loc:RecAct2+=1

Display(?Loc:RecAct2)

End

End

Loc:RegCom4+=1

Display(?Loc:RegCom4)

End

Flush(ELRTCC)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!! Proceso que generara los datos

# SR\_Generar\_Pag Routine

Data

Loc:PrcUti Decimal(6,2)

Lof:Fecha Long

Code

!! Comisiones por Vendedor

! Close(Pagos)

! Create(Pagos)

! if errorcode() then

! Loc:Mensaje\_Transaccion = 'No se Puede Crear el Archivo de Pagos por Vendedor. |Error: ' & error()

! Loc:Error\_Transaccion = true

! exit

! end

! open(Pagos, Readwrite+denyall)

! if errorcode()

! Loc:Mensaje\_Transaccion = 'No se Puede Abrir el Archivo de Pagos por Vendedor. |Error: ' & error()

! Loc:Error\_Transaccion = true

! exit

! end

!!! Comenzar Proceso Recorrer PAgos

!?Group1 {Prop:text}= 'Proceso 1: Buscando Utilidad De Pagos->En Proceso...'

!Display(?Group1)

Stream(Pagos)

Lof:Fecha= Date(CTRL:Cie\_MesActual,1,CTRL:Cie\_AnoActual)

Clear(ADXC:Record)

ADXC:FcRegistro= Lof:Fecha

SET(ADXC:LLS\_FcRegistro\_NCte,ADXC:LLS\_FcRegistro\_NCte )

Loop

NEXT(ELADXC)

If ErrorCode() Then break.

! If ADXC:CDoc <> 'P' Then Cycle. !! Se valida la misma condición que Com\_Vtas para asegurar congruencia en la información

If ~((CLIP(ADXC:Cdoc) = 'P' AND ADXC:CConcepto = 15) OR (CLIP(ADXC:Cdoc) = 'SFC' AND ADXC:CConcepto = 19)) Then Cycle.

If Month(ADXC:FcRegistro) <> CTRL:Cie\_MesActual Or |

year(ADXC:FcRegistro) <> CTRL:Cie\_AnoActual Then Break.

Loc:RegLei+=1

Display(?Loc:RegLei)

!! Sacar utlidad de la Factura

CLEAR(FAC:Record)

FAC:LFolio = ADXC:ALFolio

FAC:NFolio = ADXC:ANFolio

GET(ElFac, FAC:LLP\_LFolio\_NFolio)

If ErrorCode() Then Cycle.

If FAC:Estado = 0 Then Cycle. !! No Tomar Facturas Canceladas

IF FAC:Bnd\_AntFac then cycle. !! No tomar Anticipos

!Utilidad

Do SR\_Utilidad\_Pag

End

Flush(Pagos)

!?Group1 {Prop:text}= 'Proceso 1: Buscando Utilidad de Pagos->Terminado'

!Display(?Group1)

!?Group1:2 {Prop:text}= 'Proceso 2: Calculando Prc. De Comisión->En Proceso...'

!Display(?Group1)

Loc:RegVen=0

Loc:VenLei=0

Display

Stream(ELVEN)

Clear(Ven:Record)

SET(VEN:LLP\_NVendedor,VEN:LLP\_NVendedor)

Loop

Next(ELVen)

If ErrorCode() Then Break.

Loc:VenLei +=1

Display(?Loc:VenLei)

Loc:Pagado = 0

Loc:Utilidad = 0

Clear(TPA:Record)

TPA:NVen = VEN:NVen

SET(TPA:LLP\_NVen,TPA:LLP\_NVen)

Loop

Next(Pagos)

If ErrorCode() Then Break.

If TPA:NVen<> VEN:NVen Then Break.

Loc:Pagado += TPA:Importe

Loc:Utilidad += TPA:Utilidad

End

!! Calcular Porcentaje de utilidad

Loc:PrcUti= (Loc:Utilidad/Loc:Pagado)\*100

!! El porcentaje de comision siempre estara entre 5 y 10 %

If Loc:PrcUti < 5 Then

Loc:PrcUti= .5

End

If Loc:PrcUti > 20 Then

Loc:PrcUti= 2.0

End

If Loc:PrcUti >=5 and Loc:PrcUti < 10 Then

Loc:PrcUti = (Loc:PrcUti / 11.11)

End

If Loc:PrcUti >=10 And Loc:PrcUti <= 20 Then

Loc:PrcUti= Loc:PrcUti / 10

End

VEN:Prc\_ComN = Loc:PrcUti

VEN:Prc\_ComCN = VEN:Prc\_ComN + .5

Put(ELVEN)

If ~ErrorCode()

Loc:RegVen+=1

Display

End

End

FLUSH(ElVen)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Rutina para la utilidad

# Sr\_Utilidad\_Pag Routine

data

Lof:PNeto Decimal(12,2)

Lof:CNeto Decimal(12,2)

Code

!! Recorrer los detalles de la factura para calcular un % de utilidad General

Lof:PNeto = 0

Lof:CNeto = 0

Clear(DFAC:Record)

DFAC:Consecutivo = FAC:Consecutivo

SET(DFAC:LLP\_Consecutivo,DFAC:LLP\_Consecutivo)

Loop

Next(ELDFAC)

If ErrorCode() Then Break.

if DFAC:Consecutivo<>FAC:Consecutivo Then Break.

Lof:PNeto+=DFAC:PrecioNeto - ((DFAC:PrDescto\_NC / 100) \* DFAC:PrecioNeto)

Lof:CNeto+=(DFAC:CostoNetoVenta \* DFAC:Cantidad)

End

!! Agregar a Tabla

Clear(TPA:Record)

TPA:LFolioF = FAC:LFolio

TPA:NFolioF = FAC:NFolio

TPA:LFolioP = ADXC:LFolio

TPA:NFolioP = ADXC:NFolio

TPA:FcFecha = FAC:FcFactura

TPA:NVen = FAC:NVen

TPA:Importe = ADXC:ImpAplicar/ (1 + (CTRL:Grl\_IVA / 100))

TPA:PrecNeto = Lof:PNeto

TPA:CostNeto = Lof:CNeto

!Message(Lof:PNeto&'/'& ADXC:ImpAplicar&'= |'&Lof:PNeto/ ADXC:ImpAplicar)

! TPA:Utilidad = (Lof:PNeto-Lof:CNeto)\*(TPA:Importe/Lof:PNeto)

TPA:Utilidad = (Lof:PNeto-Lof:CNeto)\*(TPA:Importe/Lof:PNeto)

ADD(PAGOS)

If ~ErrorCode()

Loc:RegPag+=1

Display(?Loc:RegPag)

End

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!! Proceso de Llenado de Archivos

# SR\_ComMostrador Routine

Data

LocF:Partidas Long !! Partidas de la Factura.

LocF:Unidades Decimal(12,2) !! Cantidades de la Factura.

LocF:PrecFac Decimal(12,2)

LocF:PrecioNeto Decimal(12,2)

LocF:ImpDescto Decimal(12,2)

LocF:Importe1 Decimal(12,2)

!LocF:ImpDescto Decimal(12,2)

LocF:Subtotal Decimal(12,2)

Locf:TUtilidad Decimal(12,2)

Locf:Utilidad Decimal(12,2)

Locf:Ciudad Like(Ven:Ciudad)

Code

!! Rutina que abrira las tablas

!Do SR\_abrirT\_Most

!If Loc:Error = True Then Exit.

Stream(ELMVM)

Stream(ELMVMC)

LocP:FacProc =0

LocP:CapProc =0

LocP:CiuProc =0

!! Recorremos las Facturas

Clear(Fac:Record)

FAC:FcFactura=Date(CTRL:Cie\_MesActual,1,CTRL:Cie\_AnoActual)

SET(FAC:LLS\_FcFact,FAC:LLS\_FcFact)

Loop

!! 1 Sig. Registro

!! 2 Caso de Error

!! 3 Si esta Cancelada la Factura

Next(ELFAC)

If ErrorCode() Then Break.

!If FAC:FcCanc Then Cycle.

If FAC:Estado = 0 Then Cycle.

!if FAC:Bnd\_AntFac then cycle.

If Month(FAC:FcFactura) <> CTRL:Cie\_MesActual Or |

year(FAC:FcFactura) <> CTRL:Cie\_AnoActual Then Break.

LocP:FacProc +=1

Display(?LocP:FacProc)

!! Buscar Datos

!1 Clientes

!2 Capturista

!3 Vendedores

!4 Articulos

!5 Almacén

!6 Añadir a Base de datos externa

!7 Sacamos el total de partidas y cantidades por factura recorriendo los detalles

CLEAR(CTE:Record)

CTE:NCte = FAC:NCte

GET(ELCTE, CTE:LLP\_NCliente)

If ErrorCode() Then Clear(Cte:Record).

IF LocP:Mostrador THEN !! Filtro de Mostrador

!! Mostrador Seleccionado, Buscando Vendedor

CLEAR(VEN:Record)

VEN:NVen = FAC:NVen

GET(ELVen, VEN:LLP\_NVendedor)

If ErrorCode() Then Clear(Ven:Record).

If Instring('MOSTRADOR',VEN:Nombre,1,1)= 0 Then Cycle.

ELSE

!! Todas Seleccionado, Buscando Facturista

CLEAR(VEN:Record)

VEN:NVen = FAC:NVen\_Fac

GET(ELVen, VEN:LLP\_NVendedor)

If ErrorCode() Then Clear(Ven:Record).

If Instring('MOSTRADOR',VEN:Nombre,1,1)= 0 Then Cycle.

END

!! Se busca el almacén del vendedor/facturista

CLEAR(ALM:Record)

ALM:NAlm = VEN:NAlm

GET(ELALM,ALM:LLP\_NAlmacen)

If ErrorCode() Then Clear(Alm:Record).

Loc:Ciudad\_Ven = Alm:Ciudad

!! Buscamos Tipo de Vendedor/Capturista

CLEAR(VEN:Record)

VEN:NVen = FAC:NVen\_Fac

GET(ELVen, VEN:LLP\_NVendedor)

If ErrorCode() Then Clear(Ven:Record).

If VEN:Tipo\_Vend = 1 Then Cycle.

!! Se busca almacén

CLEAR(ALM:Record)

ALM:NAlm = VEN:NAlm

GET(ELALM,ALM:LLP\_NAlmacen)

If ErrorCode() Then Clear(Alm:Record).

!!! No se tomaran las Facturas de Otras ciudades

If Loc:Ciudad\_Ven <> Alm:Ciudad Then Cycle.

!! Iniciar Variables

Locf:Partidas = 0

LocF:PrecioNeto = 0

Locf:Utilidad = 0

Locf:TUtilidad = 0

!! Recorremos los detalles de la factura para

!! Tomar el numero de partidas y la cantidad de unidades

DFAC:Consecutivo = FAC:Consecutivo

SET(DFAC:LLP\_Consecutivo,DFAC:LLP\_Consecutivo)

Loop

! 1 Sig. Registro

! 2 Caso de Error

! 3 Cambio de Concecutivo

! 4 Sumatoria de partidas y unidades

Next(ElDFac)

If ErrorCode() Then Break.

If DFAC:Consecutivo<>FAC:Consecutivo Then Break.

LocF:Partidas += 1

LocF:Unidades += DFAC:Cantidad

!1-3. calcular importes de venta por detalle

LocF:PrecFac = DFAC:PrcLista\* (1 - (DFAC:PrDescuento / 100))

LocF:PrecioNeto = DFAC:Cantidad\*Locf:PrecFac

LocF:ImpDescto = (DFAC:Cantidad\*DFAC:PrcLista)- LocF:PrecioNeto

LocF:Importe1 = LocF:PrecioNeto\*(FAC:PrDescuento/100)

LocF:ImpDescto += LocF:Importe1

LocF:Subtotal = LocF:PrecioNeto

!! calculo de Utilidad

!Lof:PrecVtaNeto = DFAC:PrecioNeto - ((DFAC:PrDescto\_NC / 100) \* DFAC:PrecioNeto)

Locf:Utilidad = LocF:PrecioNeto - (DFAC:CostoNetoVenta \* DFAC:Cantidad)

Locf:TUtilidad += Locf:Utilidad

End

!! Llenamos tabla Requerida para el Calculo

MVM:NVen\_Fac = FAC:NVen\_Fac

GET(ELMVM,MVM:LLP\_Ven\_Fac )

If Errorcode()

Clear(MVM:Record)

MVM:NVen\_Fac = FAC:NVen\_Fac

MVM:NumFacts = 1

MVM:Partidas = Locf:Partidas

! Capturista

Clear(TVN:Record)

TVN:NVen = FAC:NVen\_Fac

TVN:Tipo = 'A'

TVN:Nivel = 'F'

TVN:Valor = ''

GET(ArcVtasNtas,TVN:LLP\_Id)

If ErrorCode() Then Clear(TExp:Record).

MVM:Venta = TVN:Venta

MVM:Ciudad = Alm:Ciudad

ADD(ELMVM)

If ~ErrorCode()

LocP:CapProc +=1

Display(?LocP:CapProc)

End

Else

MVM:NumFacts += 1

MVM:Partidas += Locf:Partidas

PUT(ELMVM)

If ErrorCode()

!! Nada Aun

End

End

!! Llenamos tabla Requerida para el Calculo

MVMC:Ciudad = Alm:Ciudad

GET(ELMVMC,MVMC:LLP\_Ciudad )

If Errorcode()

Clear(MVMC:Record)

MVMC:Ciudad = Alm:Ciudad

MVMC:NumFacts = 1

MVMC:Partidas = Locf:Partidas

!!! Buscamos Ventas Netas

Clear(TVNC:Record)

TVNC:Ciudad = Alm:Ciudad

TVNC:Tipo = 'A'

TVNC:Nivel = 'C'

TVNC:Valor = ''

GET(ArcVtasNtasC, TVNC:LLP\_Id)

If ErrorCode() Then Clear(TVNC:Record).

MVMC:Venta = TVNC:Venta

!!! Buscamos Ventas Netas

Clear(TVNC:Record)

TVNC:Ciudad = Alm:Ciudad

TVNC:Tipo = 'C'

TVNC:Nivel = 'C'

TVNC:Valor = ''

GET(ArcVtasNtasC, TVNC:LLP\_Id)

If ErrorCode() Then Clear(TVNC:Record).

MVMC:Utilidad = TVNC:Venta

ADD(ELMVMC)

If ~ErrorCode()

LocP:CiuProc +=1

Display(?LocP:CiuProc)

End

Else

MVMC:NumFacts += 1

MVMC:Partidas += Locf:Partidas

!MVMC:Venta += FAC:SubTotal

!MVMC:Utilidad += Lof:TUtilidad

PUT(ELMVMC)

If ErrorCode()

!! Nada Aun

End

End

End

Flush(ELMVM)

Flush(ELMVMC)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Proceso de agrupacion de cotizaciones por vendedor CAT

# Sr\_CotVen Routine

data

LocF:Partidas long

code

!Inicializar

LocP:CotProc = 0

!Recorrer las cotizaciones

clear(Cot:Record)

COT:FcDocto=Date(CTRL:Cie\_MesActual,1,CTRL:Cie\_AnoActual)

SET(COT:LLS\_FcDocto,COT:LLS\_FcDocto)

Loop

Next(ELCOT)

If ErrorCode() then break.

If Month(COT:FcDocto)<> CTRL:Cie\_MesActual Or |

year(COT:FcDocto) <> CTRL:Cie\_AnoActual Then Break.

If COT:Estado <> 1 then cycle.

!Buscar vendedor

clear(Ven:Record)

VEN:NVen = COT:NVen\_Fac

get(ELVEN, VEN:LLP\_NVendedor)

If ErrorCode() then cycle.

If VEN:Tipo\_Vend <> 3 then cycle.

!Buscar Almacen

CLEAR(ALM:Record)

ALM:NAlm = VEN:NAlm

GET(ELALM,ALM:LLP\_NAlmacen)

If ErrorCode() Then cycle.

!Inizialiar

LocF:Partidas = 0

LocP:CotProc += 1

display(?LocP:CotProc)

!Contabilizar los detalles

clear(DCot:Record)

DCOT:LFolio = COT:LFolio

DCOT:NFolio = COT:NFolio

set(DCOT:LLP\_Folio\_NPartida, DCOT:LLP\_Folio\_NPartida)

Loop

next(ELDCOT)

If ErrorCode() then break.

If DCOT:LFolio <> COT:LFolio or |

DCOT:NFolio <> COT:NFolio then break.

LocF:Partidas += 1

End

! Guardar en tabla temporal el numero de cotizaciones por vendedor CAT

clear(CV1:Record)

CV1:NVen\_Fac = COT:NVen\_Fac

get(COTVEN, CV1:LLP\_Ven\_Fac)

If ErrorCode() then

CV1:Ciudad = ALM:Ciudad

CV1:Partidas = LocF:Partidas

Add(COTVEN)

Else

CV1:Partidas += LocF:Partidas

put(COTVEN)

End

End

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!