

Documentacion - SQL

ROLLBACK

1. Al comienzo de cada cambio que se desea realizar, se debe ejecutar:

```
BEGIN TRANSACTION
```

Esto permite al SQL Server reconocer el inicio de una serie de ejecuciones que NO queremos que se confirmen.

2. Realizar los cambios que queramos hacer, como por ejemplo:

```
UPDATE ERP_WEB_DINAMICAS SET TIPO_RELACION_ARTICULOS = 'EX' WHERE ID_DINAMICA = 247
```

NOTA: Si despues queremos ver nuestros cambios NO confirmados debemos ejecutar una consulta del estilo

```
SELECT * FROM ERP_WEB_DINAMICAS WITH(NOLOCK) WHERE ID_DINAMICA > 244
```

3. Si queremos dar el OK del cambio ejecutamos:

```
COMMIT
```

Si queremos volver para atrás ejecutamos:

```
ROLLBACK
```

```
adm_buscar_string 'full_PBI_Ventas_union'
```

```
Sp_helpfull_PBI_Ventas_union
```

Venta Mayorista

Todo esta dentro de la base de SQL de factu_full_reportes

Vista - full_Proveedores_Incidencia_ventas

- Código @Augusto

```
select
itm.ARTICULO,art.DESCRIP_ARTI,abs(sum(itm.CANT)) as
venta_total,comp.CLIENTE,cli.NOM_FANTASIA,cli.AGRU_3 ,Dateadd(Month, Datediff(Month, 0, DATEADD(m,
-6,current_timestamp)), 0) as fecha_desde
from factu_full_central.dbo.COMP_EMITIDOS as comp
inner join factu_full_central.dbo.RENG_FAC as itm on (comp.TIPO=itm.TIPO_FACT and
comp.NUM=itm.NUM_FACT)
inner join factu_full_central.dbo.CLIENTES as cli on comp.CLIENTE=cli.NUM_CLIENTE
inner join factu_full_central.dbo.ARTICULOS as art on itm.ARTICULO=art.COD_ARTICULO
where comp.FECHA >= Dateadd(Month, Datediff(Month, 0, DATEADD(m, -6,current_timestamp)), 0)
and cli.AGRU_3 in ('FULL','PRO')
and comp.ACTUA_STOCK = 'S'
and USA_COMPO = 'N'
and art.SIN_STOCK = 'N'
group by itm.ARTICULO,art.DESCRIP_ARTI,comp.CLIENTE,cli.AGRU_3,cli.NOM_FANTASIA
```

ARTICULO	DESCRIP_ARTI	venta_total	CLIENTE	NOM_FANTASIA	AGRU_3	fecha_desde
C1	Campari - 750 ml	48.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C100	Chandon Brut Nature - 750 ml	3.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C101	Chandon Rose - 750 ml	12.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C102	Navarro Correas Extra Brut - 750 ml	24.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C103	Mumm Extra Brut - 750 ml	24.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C104	Baron B Extra Brut - 750 ml	12.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C105	Baron B Rose - 750 ml	6.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C123	Cafayate Malbec - 750 ml	72.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000
C125	Luis M. Malbec - 750 ml	24.0000	1005	Fullescabio Alem	FULL	2022-11-01 00:00:00.000

La vista busca desde en la base de Central todas las ventas(con movimiento de stock) de los últimos 6 meses para las siguientes categorías de clientes (AGRU_3)

- FULL = Sucursales Fullescabio
- PRO = Sucursales Fullescabio Propias

Se toman todos los movimientos que afectaron el stock, se limitó el listado de artículos para no contar los que tienen componentes (PR,B) y los que son de tipo servicio (A)

SP - full_Proveedores_articulos_incidencia

V1

.

Código [@Hernan Reta](#)

```
@Cod_articulo varchar(10),
@cantidad int
as

SELECT a.articulo, a.descrip_arti, a.cliente, venta_total / cantidad as Incidencia,
round(venta_total / cantidad * @cantidad,0) as Cantidad_Final
FROM full_Proveedores_Incidencia_ventas as a left join (SELECT b.articulo as articulo,
sum(b.venta_total) as cantidad
FROM full_Proveedores_Incidencia_ventas b
where b.articulo = @Cod_articulo
group by b.articulo ) as T
on a.articulo = t.articulo
where a.articulo = @Cod_articulo
```

Se envía el SKU y la cantidad de unidades que necesitamos repartir entre todas las sucursales, el sql multiplica el porcentaje de venta por la cantidad ingresada y redondea el numero para dar la cantidad final por cliente.

La suma de las cantidades finales no siempre es el mismo que la cantidad cargada inicialmente

V2

.

Código [@Augusto](#)

```
@Cod_articulo varchar(10) ,
@cantidad int ,
@comprobante int
as
```

```

;

-----CALCULO DE CANTIDADES FINALES-----
declare @numeracion int =(select isnull(max(nro_ejecucion)+1,1) from proveedores_incidencia);

with TEMP as (
SELECT a.articulo, a.descrip_arti, a.cliente, venta_total / cantidad as Incidencia,
round(venta_total / cantidad * @cantidad,0) as Cantidad_Final
FROM full_Proveedores_Incidencia_ventas as a
left join (SELECT b.articulo as articulo, sum(b.venta_total) as cantidad
FROM full_Proveedores_Incidencia_ventas b
where b.articulo = @Cod_articulo
group by b.articulo ) as T
on a.articulo = t.articulo
where a.articulo = @Cod_articulo
)

-----CARGA A LA BASE FINAL-----
insert into
dbo.proveedores_incidencia(NRO_FILA,articulo,descrip_arti,cliente,Incidencia,Cantidad_final,nro_ejecucion
,cantidad_ejecucion
,cargado
,fecha_ejecucion
,nro_comprobante
,precio_uni)
select row_number() OVER (ORDER BY CANTIDAD_FINAL ASC) AS NRO_FILA
,trim(TEMP.articulo),trim(TEMP.descrip_arti)

,TEMP.cliente,TEMP.Incidencia,TEMP.Cantidad_final,@numeracion,@cantidad,trim('N'),getdate(),@compr
obante
,LIS.PRECIO_VTA
from TEMP
LEFT JOIN factu_full_central.DBO.LISTAS_ITEMS AS LIS ON TEMP.ARTICULO=LIS.ARTICULO AND
LIS.LISTA_CODI = 'LF'

-----CONTROL DE RESULTADO FINAL-----
declare @final int,
@cant_sp int,
@control int
;

select @final = sum(cantidad_final) from proveedores_incidencia
where articulo = @Cod_articulo and nro_ejecucion = @numeracion
group by articulo,nro_ejecucion,cantidad_ejecucion

select @cant_sp = cantidad_ejecucion from proveedores_incidencia
where articulo = @Cod_articulo and nro_ejecucion = @numeracion

select @control = @cant_sp - @final

```

-----MODIFICACION AL ARTICULO CON MAS CANTIDAD-----

```
-----  
if @control <> 0  
begin  
    update proveedores_incidencia set cantidad_final = (cantidad_final + @control) where articulo  
= @Cod_articulo and nro_ejecucion = @numeracion and  
    NRO_FILA = (select top 1 NRO_FILA from proveedores_incidencia where articulo = @Cod_articulo  
and nro_ejecucion = @numeracion ORDER BY cantidad_final DESC)  
end  
  
  
@Cod_articulo varchar(10) ,  
@cantidad int ,  
@comprobante int  
as;  
-----CALCULO DE CANTIDADES FINALES-----  
declare @numeracion int =(select isnull(max(nro_ejecucion)+1,1) from proveedores_incidencia);  
  
with TEMP as (  
SELECT a.articulo, a.descrip_arti, a.cliente, venta_total / cantidad as Incidencia,  
round(venta_total / cantidad * @cantidad,0) as Cantidad_Final  
    FROM full_Proveedores_Incidencia_ventas as a left join (SELECT b.articulo as articulo,  
sum(b.venta_total) as cantidad  
        FROM full_Proveedores_Incidencia_ventas b  
        where b.articulo = @Cod_articulo  
        group by b.articulo ) as T  
        on a.articulo = t.articulo  
        where a.articulo = @Cod_articulo  
)
```

Se uso la primera parte de la version 1 para calcular los totales de cantidad final por cliente, se agrego el valor @numeracion para calcular el próximo numero de ejecución que tiene que grabar en la tabla

-----CARGA A LA BASE FINAL-----

```
insert into  
dbo.proveedores_incidencia(NRO_FILA,articulo,descrip_arti,cliente,Incidencia,Cantidad_final,nro_ejecucion  
,cantidad_ejecucion  
,cargado  
,fecha_ejecucion  
,nro_comprobante  
,precio_uni)  
select row_number() OVER (ORDER BY CANTIDAD_FINAL ASC) AS NRO_FILA  
,trim(TEMP.articulo),trim(TEMP.descrip_arti)
```

```

, TEMP.cliente, TEMP.Incidencia, TEMP.Cantidad_final, @numeracion, @cantidad, trim('N'), getdate(), @comprobante
, LIS.PRECIO_VTA
from TEMP
LEFT JOIN factu_full_central.DBO.LISTAS_ITEMS AS LIS ON TEMP.ARTICULO=LIS.ARTICULO AND LIS.LISTA_CODI =
'LF'

```

Se inserta los resultados de la primera consulta dentro de la tabla [proveedores_incidencia](#), dentro de este paso se calcula los números de filas (la numeración de la fila corresponde a la cantidad final, a mayor cantidad mayor numero de fila). También se agrega la numeración y la fecha de ejecución para poder identificarlo, junto con el campo @cantidad para revisar el calculo de ser necesario.

```

-----CONTROL DE RESULTADO FINAL-----
declare @final int,
        @cant_sp int,
        @control int
;

select @final = sum(cantidad_final) from proveedores_incidencia
where articulo = @Cod_articulo and nro_ejecucion = @numeracion
group by articulo,nro_ejecucion,cantidad_ejecucion

select @control = @cantidad - @final

```

Controlamos el resultado final para que siempre corresponda con la cantidad cargada inicialmente, el campo @final contiene la suma de todos las cantidades que genero al principio. El campo control resta la cantidad cargada al principio y la cantidad final, acá vemos si calculo bien

```

-----MODIFICACION AL ARTCILULO CON MAS CANTIDAD-----
if @control <> 0
begin
    update proveedores_incidencia set cantidad_final = (cantidad_final + @control) where articulo =
@Cod_articulo and nro_ejecucion = @numeracion and
    NRO_FILA = (select top 1 NRO_FILA from proveedores_incidencia where articulo = @Cod_articulo and
    nro_ejecucion = @numeracion ORDER BY cantidad_final DESC)
end

```

Si el resultado del control es diferente a 0 se modifica el campo cantidad_final con la cantidad mas grande de toda esa ejecución, usamos el campo NRO_FILA para buscar el numero mas grande. No buscamos el numero mas grande directamente porque en el caso que el numero mayor se repita en la ejecución el sistema modifica los dos números.

```

USE [factu_full_reportes]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[full_Proveedores_articulos_incidencia]
        @Cod_articulo = N'C145',
        @cantidad = 100,
        @comprobante = 1

SELECT  'Return Value' = @return_value

GO

```

Para ejecutar el sp usamos el siguiente query, se pasan los valores de sku y cantidad. El valor @comprobante se usa para la creacion de las facturas

V3

- Código [@Augusto](#)

```

@Cod_articulo varchar(10) ,
@cantidad int ,
@comprobante int
as
;
declare @numeracion int =(select isnull(max(nro_ejecucion)+1,1) from proveedores_incidencia);

declare @TEMP2 TABLE  (COD_ARTICULO varchar(80),EN_VISTA varchar(80),DESCRIP VARCHAR (330)
,CLIENTE INT
,NOMBRE_FANTASIA VARCHAR(80)
,CANTIDAD NUMERIC (12, 4)
)

-----CHECK DE ARTICULOS PARA EL CALCULO-----
insert into @TEMP2(COD_ARTICULO,EN_VISTA,DESCRIP,CANTIDAD,CLIENTE,NOMBRE_FANTASIA)
select distinct COD_ARTICULO,isnull(ARTICULO,'NO') as EN_VISTA,ART.DESCRIPT_ARTI,
CAST(ABS(CHECKSUM(NEWID())) % (25 -1) AS INT) + 1 as cantidad
,CLI.NUM_CLIENTE,CLI.NOM_FANTASIA
from factu_full_central.dbo.ARTICULOS as art
left join full_Proveedores_Incidencia_ventas as piv on art.COD_ARTICULO=piv.ARTICULO
,factu_full_central.dbo.clientes AS CLI
where CLI.AGRU_3 in ('PRO')
AND ART.COD_ARTICULO = @Cod_articulo

declare @en_vista varchar(10)

```

```

select @en_vista = en_vista from @TEMP2

if @Cod_articulo = @en_vista
begin
-----SI EL ARTICULO ESTA DENTRO DE LA VISTA-----
with TEMP as (
SELECT a.articulo, a.descrip_arti, a.cliente, venta_total / cantidad as Incidencia, round(venta_total / cantidad * @cantidad,0) as Cantidad_Final
FROM full_Proveedores_Incidencia_ventas as a
left join (SELECT b.articulo as articulo, sum(b.venta_total) as cantidad
          FROM full_Proveedores_Incidencia_ventas b
          where b.articulo = @Cod_articulo
          group by b.articulo ) as T
on a.articulo = t.articulo
where a.articulo = @Cod_articulo
)
-----CARGA A LA BASE FINAL-----
insert into
dbo.proveedores_incidencia(NRO_FILA,articulo,descrip_arti,cliente,Icidencia,Cantidad_final,nro_ejecucion
,cantidad_ejecucion
,cargado
,fecha_ejecucion
,nro_comprobante
,precio_uni)
select row_number() OVER (ORDER BY CANTIDAD_FINAL ASC) AS NRO_FILA
,trim(TEMP.articulo),trim(TEMP.descrip_arti)

,TEMP.cliente,TEMP.Icidencia,TEMP.Cantidad_final,@numeracion,@cantidad,trim('N'),getdate(),@comprobante
,LIS.PRECIO_VTA
from TEMP
LEFT JOIN factu_full_central.DBO.LISTAS_ITEMS AS LIS ON TEMP.ARTICULO=LIS.ARTICULO AND LIS.LISTA_CODI =
'LF'

end
else
begin
-----SI EL ARTICULO NO ESTA EN LA VISTA-----
insert into dbo.proveedores_incidencia(NRO_FILA,articulo,descrip_arti,cliente,
Icidencia,Cantidad_final,nro_ejecucion
,cantidad_ejecucion
,cargado
,fecha_ejecucion
,nro_comprobante
,precio_uni)
select row_number() OVER (ORDER BY T.CANT_T ASC) AS NRO_FILA,TE.COD_ARTICULO,TE.DESCRIP,TE.CLIENTE
,TE.CANTIDAD / T.CANT_T AS INCIDENCIA,round(TE.CANTIDAD / CANT_T* @cantidad,0) AS
CANTIDAD_FINAL,@NUMERACION,
@cantidad,'N',GETDATE(),@comprobante,lis.PRECIO_VTA

```

```

FROM @TEMP2 AS TE
LEFT JOIN (SELECT COD_ARTICULO,SUM(CANTIDAD) AS CANT_T FROM @TEMP2 GROUP BY COD_ARTICULO)
AS T ON TE.COD_ARTICULO=T.COD_ARTICULO
LEFT JOIN factu_full_central.DBO.LISTAS_ITEMS AS LIS ON te.COD_ARTICULO collate
database_default=lis.ARTICULO AND LIS.LISTA_CODI = 'LF'
end

declare @final int,
        @cant_sp int,
        @control int
;

select @final = sum(cantidad_final) from proveedores_incidencia
where articulo = @Cod_articulo and nro_ejecucion = @numeracion
group by articulo,nro_ejecucion,cantidad_ejecucion

select @cant_sp = cantidad_ejecucion from proveedores_incidencia
where articulo = @Cod_articulo and nro_ejecucion = @numeracion

select @control = @cant_sp - @final

-----MODIFICACION AL ARTCILULO CON MAS CANTIDAD-----
if @control <> 0
begin
    update proveedores_incidencia set cantidad_final = (cantidad_final + @control) where articulo =
@Cod_articulo and nro_ejecucion = @numeracion and
NRO_FILA = (select top 1 NRO_FILA from proveedores_incidencia where articulo = @Cod_articulo and
nro_ejecucion = @numeracion ORDER BY cantidad_final DESC)
end

```

- cambios

- Tiene que tomar el nro de comprobante desde el tipo de tx que armemos
- falta grabar el precio unitario neto de iva
-

SP -

full_Proveedores_articulos_incidencia_fecha

.

Código

```
@fecha date ,
@ejecucion int
as
;

DECLARE @SKU VARCHAR(10),
@CANT INT;

DECLARE @TEMP TABLE (FACT VARCHAR(10),SKU VARCHAR(10), CANT INT, PRECIO INT);

INSERT INTO @TEMP(FACT,SKU,CANT,PRECIO)
select CONCAT(COMP.TIPO,'-',COMP.NUM) AS FACT,ITM.ARTICULO AS SKU,ABS(ITM.CANT) AS
CANT,ITM.PRECIO_NETO*1.21 AS PRECIO
from FACTU_FULL_CENTRAL.DBO.COMP_EMITIDOS AS COMP
INNER JOIN FACTU_FULL_CENTRAL.DBO.RENG_FAC AS ITM ON COMP.TIPO=ITM.TIPO_FACT AND
COMP.NUM=ITM.NUM_FACT
INNER JOIN FACTU_FULL_CENTRAL.DBO.CLIENTES AS CLI ON COMP.CLIENTE=CLI.NUM_CLIENTE
WHERE COMP.FECHA = @fecha
AND CLI.AGRU_3 = 'MAYOR'
AND COMP.ACTUA_STOCK = 'S'
AND LEFT(COMP.TIPO,1) = 'F'
and left(itm.articulo,1) in ('C','P')
and itm.COMPO_ITEM is null

DECLARE @RowCount INT = (SELECT COUNT(*) FROM @TEMP);

WHILE @RowCount > 0 BEGIN

SELECT @SKU= SKU,@CANT= CANT FROM @TEMP
ORDER BY SKU DESC OFFSET @RowCount - 1 ROWS FETCH NEXT 1 ROWS ONLY;

EXEC full_Proveedores_articulos_incidencia @SKU,@CANT,@ejecucion

SET @RowCount -= 1;
END
```

```

@fecha date ,
@ejecucion int
as
;

DECLARE @SKU VARCHAR(10),
@CANT INT;

DECLARE @TEMP TABLE (FACT VARCHAR(10),SKU VARCHAR(10), CANT INT, PRECIO INT);

INSERT INTO @TEMP(FACT,SKU,CANT,PRECIO)
select CONCAT(COMP.TIPO,'-',COMP.NUM) AS FACT,ITM.ARTICULO AS SKU,ABS(ITM.CANT) AS
CANT,ITM.PRECIO_NETO*1.21 AS PRECIO
from FACTU_FULL_CENTRAL.DBO.COMP_EMITIDOS AS COMP
INNER JOIN FACTU_FULL_CENTRAL.DBO.RENG_FAC AS ITM ON COMP.TIPO=ITM.TIPO_FACT AND COMP.NUM=ITM.NUM_FACT
INNER JOIN FACTU_FULL_CENTRAL.DBO.CLIENTES AS CLI ON COMP.CLIENTE=CLI.NUM_CLIENTE
WHERE COMP.FECHA = @fecha
AND CLI.AGRU_3 = 'MAYOR'
AND COMP.ACTUA_STOCK = 'S'
AND LEFT(COMP.TIPO,1) = 'F'
and left(itm.articulo,1) in ('C','P')
and itm.COMPO_ITEM is null

```

Se declaran las variables, se crea una variable de tipo de tabla llamada @TEMP . En esta tabla se ingresan los valores de numero de factura, sku, cantidad y precio de todas las facturas que movieron stock de los clientes mayoristas en un día en específico.

```

DECLARE @RowCount INT = (SELECT COUNT(*) FROM @TEMP);

WHILE @RowCount > 0 BEGIN

SELECT @SKU= SKU,@CANT= CANT FROM @TEMP
ORDER BY SKU DESC OFFSET @RowCount - 1 ROWS FETCH NEXT 1 ROWS ONLY;

EXEC full_Proveedores_articulos_incidencia @SKU,@CANT,@ejecucion

SET @RowCount -= 1;
END

```

En esta parte se crea una variable @rowcount donde se declara la cantidad de líneas que conforman la tabla @temp. El conteo de filas se utiliza para correr todos los sku y se envían el sku y cantidad hacia el store procedure Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6444>)

Tabla - proveedores_incidencia

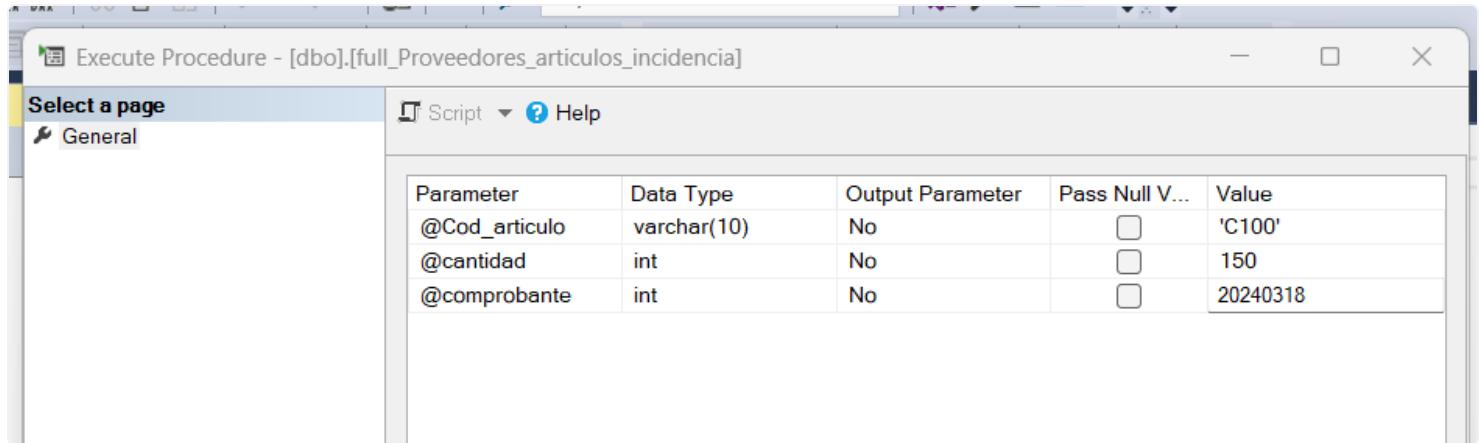
ARTIC...	DESCRIP_ARTI	CLIENTE	INCIDENCIA	CANTIDAD...	NRO_EJECUCION	CANTIDAD_EJ...	CARGADO	FECHA_EJECUCI...	NRO_FILA	NRO_CO...	PRECIO_UNI
3 C8665	Andes Origen Rubia ...	7044	0.0276	6.0000	5.0000	200.0000	N	2023-05-18 09:33...	16.0000	3	235.0000
4 C8665	Andes Origen Rubia ...	1032	0.0367	7.0000	5.0000	200.0000	N	2023-05-18 09:33...	17.0000	3	235.0000
5 C8665	Andes Origen Rubia ...	1034	0.0387	8.0000	5.0000	200.0000	N	2023-05-18 09:33...	18.0000	3	235.0000
6 C8665	Andes Origen Rubia ...	7793	0.0466	9.0000	5.0000	200.0000	N	2023-05-18 09:33...	19.0000	3	235.0000
7 C8665	Andes Origen Rubia ...	7062	0.0512	10.0000	5.0000	200.0000	N	2023-05-18 09:33...	20.0000	3	235.0000
8 C8665	Andes Origen Rubia ...	1005	0.0547	11.0000	5.0000	200.0000	N	2023-05-18 09:33...	21.0000	3	235.0000
9 C8665	Andes Origen Rubia ...	1007	0.0604	12.0000	5.0000	200.0000	N	2023-05-18 09:33...	22.0000	3	235.0000
0 C8665	Andes Origen Rubia ...	1014	0.0637	13.0000	5.0000	200.0000	N	2023-05-18 09:33...	23.0000	3	235.0000
1 C8665	Andes Origen Rubia ...	1049	0.0647	13.0000	5.0000	200.0000	N	2023-05-18 09:33...	24.0000	3	235.0000
2 C8665	Andes Origen Rubia ...	8980	0.1283	26.0000	5.0000	200.0000	N	2023-05-18 09:33...	25.0000	3	235.0000
3 C8665	Andes Origen Rubia ...	1052	0.2335	45.0000	5.0000	200.0000	N	2023-05-18 09:33...	26.0000	3	235.0000

Dentro de esta tabla se agrega el resultado de la ejecución del sp [full_proveedores_articulos_incidencia](#)

- articulo : código del artículo
- descrip_arti: descripción del artículo
- cliente: número del cliente
- incidencia: porcentaje de venta para ese artículo en los últimos 6 meses
- Cantidad_final: cantidad final repartida entre todos los clientes
- nro_ejecucion: número identificador de las ejecuciones
- cantidad_ejecucion: cantidad original cargada al momento de correr el sp
- cargado: si fue leido y cargado en una factura (N = No, S = Si)
 - al ejecutarse el sp siempre agrega el valor N
- fecha_ejecucion: fecha y hora en cuando se cargo en la tabla
- nro_fila: fila numerada por ejecucion, se ordena de mayor a menor por la cantidad final cargada en la ejecucion
- nro_comprobante: se agrega el número de comprobante que va a usar rodrigo para la facturación
- precio_uni: contiene el precio unitario del artículo para la lista LF al momento de realizar la consulta

Como hacer el ajuste manualmente

1 - Ejecutar SP full_Proveedores_articulos_incidencia en la base de reportes



```
USE [factu_full_reportes]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[full_Proveedores_articulos_incidencia]
        @Cod_articulo = N'C100',
        @cantidad = 150,
        @comprobante = 20240318

SELECT  'Return Value' = @return_value

GO
```

El resultado se busca por codigo de articulo y nro de comprobante

```
select *
from proveedores_incidencia
where ARTICULO in ( 'C100', 'C104')
and NRO_COMPROBANTE = 20240318
```

En el directorio C:\Users\HernanReta\OneDrive\Reportes a Proveedores se encuentra el file File DRW Venta mayorista .xlsx

2 - Copiar la consulta del resultado

3 - Poner Comprobante, fecha y ejecutar el SQL en la base de central

```
insert into RENG_FAC_DRAW VALUES( 'FCN',2212,'20240315',1013,1,'C100',3,6300,'DEP');
insert into RENG_FAC_DRAW VALUES( 'FCN',2213,'20240315',1018,1,'C100',3,6300,'DEP');
```

SP - Mayorista

Diagrama general de los SP'S

[MAY.drawio](#)

SP - Clientes

El buscador utiliza 3 SP's, para los caso de nombre, nombre fantasia o numero

may_client_busq_nomb - Busqueda de cliente por nombre

```
@nom varchar(50)=NULL as

select
    NUM_CLIENTE,
    RAZON,
    NOM_FANTASIA,
    L.LISTA_DESCRIP
from
    clientes as c inner join LISTAS_CABE as l on (c.LISTA_CODI = l.LISTA_CODI)
where
    ESTADO = 'A'
    AND c.NOM_FANTASIA LIKE '%' + @nom + '%' --
```

EXEC may_client_busq_nomb @nom = 'Vino'

may_client_busq_razon - Busqueda de cliente por razon

```
@razon varchar(50) as

select
    NUM_CLIENTE,
    RAZON,
    NOM_FANTASIA,
    L.LISTA_DESCRIP
from
    clientes as c inner join LISTAS_CABE as l on (c.LISTA_CODI = l.LISTA_CODI)
where
    ESTADO = 'A'
    AND c.RAZON LIKE '%' + @razon + '%' --
```

EXEC may_client_busq_razon @razon = 'Vino'

may_client_busq - Busqueda el cliente por numero

```
@num int =NULL as

select
    NUM_CLIENTE,
    RAZON,
    NOM_FANTASIA,
    L.LISTA_DESCRIP
from
    clientes as c inner join LISTAS_CABE as l on (c.LISTA_CODI = l.LISTA_CODI)
where
    ESTADO = 'A'
    and c.num_cliente = @num -- input de num de cliente
```

```
exec may_client_busq @num = '1003'
```

SP - Articulos

may_articulos

- Trae los articulos, si el código existe, trae los datos, sino, viene vacio.
- Se puede buscar por descripción - Si viene vacia la descripcion, trae todos los articulos.
- @lista_codi hace referencia a la lista de precios.

```
@cod_art varchar(30) = NULL,
@descrip varchar(50) = NULL ,
@lista_cod  varchar(10)  as;

if @descrip is not null
    select a.COD_ARTICULO,
           a.DESCRIP_ARTI,
           l.PRECIO_VTA
      from
        ARTICULOS a inner join LISTAS_ITEMS l on (a.cod_articulo = l.articulo)
     where
       a.COD_ARTICULO like 'C%'
       and a.ACTIVO = 'S'
      --and a.COD_ARTICULO = @cod_art
       and a.DESCRIP_ARTI like '%' +@descrip+'%'
       and l.LISTA_CODI = @lista_cod
  else
    select a.COD_ARTICULO,
           a.DESCRIP_ARTI,
           l.PRECIO_VTA
      from
        ARTICULOS a inner join LISTAS_ITEMS l on (a.cod_articulo = l.articulo)
     where
       a.COD_ARTICULO = @cod_art
       and a.ACTIVO = 'S'
      --and a.COD_ARTICULO like @cod_art+'%'
       and a.COD_ARTICULO like 'C%'
       and l.LISTA_CODI = @lista_cod
```

```
exec [may_articulos] @cod_art = 'C9', @lista_cod = 'HAM'
```

SP - partidas

[dbo].[may_arti_partidas]

Trae las partidas, pendientes, del articulo

@cod_art hace referencia al código de articulo por el cual se necesita detallar las partidas pendientes. **la variable cod articulo, no puede ser null**

```
@cod_art varchar(30),
@cod_depo varchar(30) = NULL as
;

if @cod_depo is not null and @cod_depo in('DEP', 'TRA')--CERRADO
    SELECT c.COD_ARTICULO AS ARTICULO, a.DESCRIP_ARTI, COD_DEPO AS DEPO, COD_PARTIDA, CANT_PEND,
    FECHA_VENCI, UBICACION_PARTIDA, COSTO_UNI

    FROM COMPRO_PARTIDAS C
        inner join ARTICULOS a on (c.COD_ARTICULO = a.COD_ARTICULO)
    WHERE
        C.COD_ARTICULO = @cod_art
        AND COD_DEPO = @cod_depo
        AND CANT_PEND > 0
else--ABIERTO/GENERAL
SELECT * from COMPRO_PARTIDAS
WHERE COD_ARTICULO = @cod_art
```

SP - Pedidos (tipo)

[dbo].[may_comp_pedidos]

Se usa en la cabecera del pedido, para señalar que tipo de comprobante va a generar en RP

- Solo trae 3 tipos de pedidos por el momento

```
as  
select COD_COMP , DESCRIP from COMP  
where COD_COMP in ( 'DDM' , 'DDW' , 'DDI' )
```

SP - Valida Excel

```
n/***** Object:  StoredProcedure [dbo].[may_valida_excel]      Script Date: 15/10/2024 23:19:53 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[may_valida_excel]
    @cod_articulo VARCHAR(50),
    @num_cliente INT
AS
BEGIN
    -- Comprobar si existe el SKU en la tabla ARTICULOS
    IF NOT EXISTS (SELECT 1 FROM ARTICULOS WHERE cod_articulo = @cod_articulo)
        BEGIN
            -- Si no existe el SKU, lanzar un error
            RAISERROR ('El SKU no existe en la base de datos.', 16, 1);
            RETURN;
        END

    -- Comprobar si existe el Número de Cliente en la tabla CLIENTES
    IF NOT EXISTS (SELECT 1 FROM CLIENTES WHERE num_cliente = @num_cliente)
        BEGIN
            -- Si no existe el Número de Cliente, lanzar un error
            RAISERROR ('El Número de Cliente no existe en la base de datos.', 16, 1);
            RETURN;
        END

    -- Si ambos existen, retornar TRUE (o lo que necesites)
    PRINT 'Ambos registros existen.';
END
```

SP - Lista de Precios

[may_lista_precios]

Listas de precios con sus códigos. Unicamente muestra 3 listas de precios.

```
select LISTA_DESCRIP from LISTAS_CABE  
where LISTA_CODI in ('HAM', 'CFM', 'LF1')
```

SP - Ventas adicionales

```
use factu_full_central_desa_nacho
```

```
/* Prueba:
```

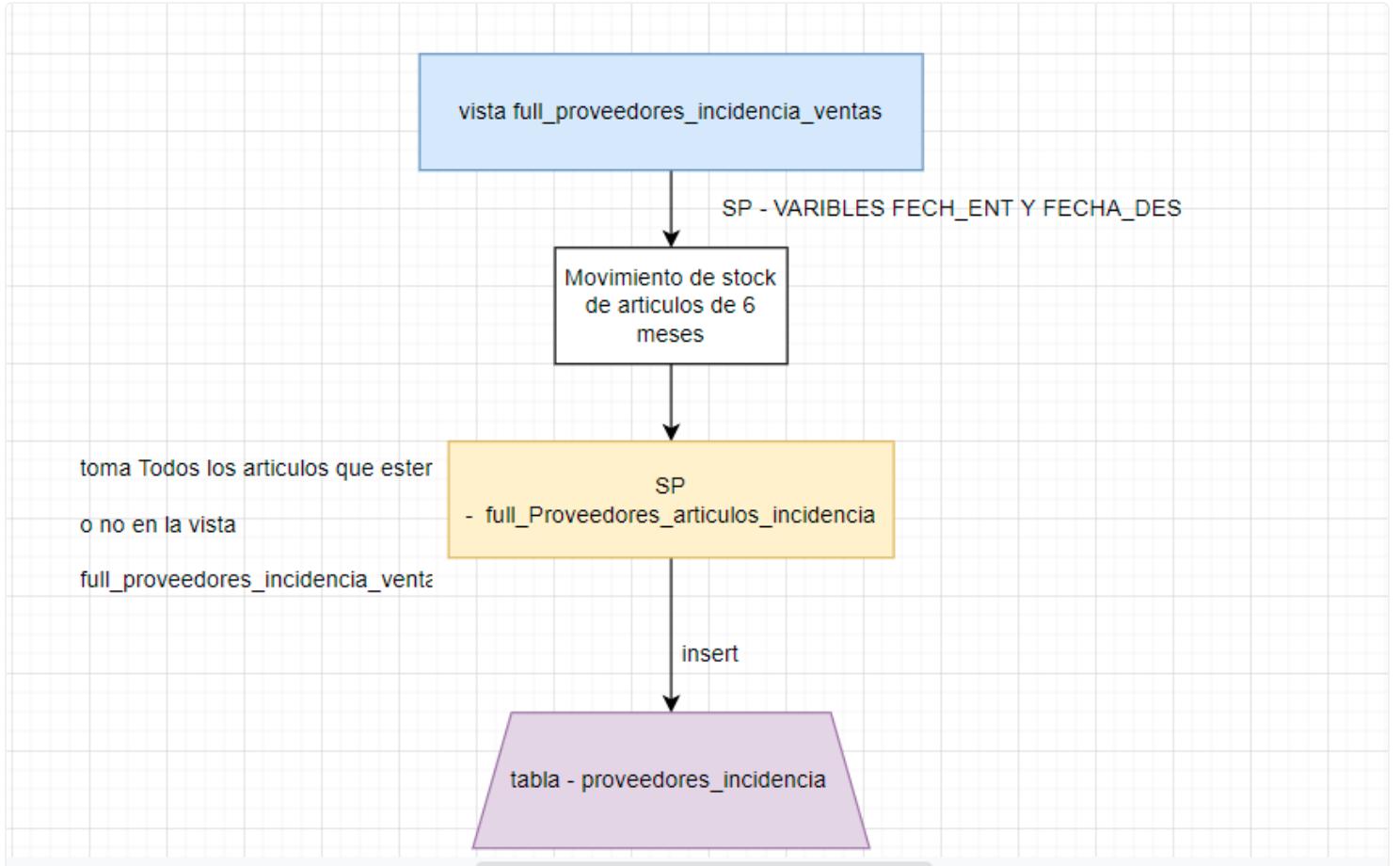
```
use factu_full_central_desa_nacho
exec [may_Proveedores_articulos_incidencia]
@Cod_articulo = 'C1' ,
@cantidad = 2000,
@comprobante = '1'
*/
```

Cree la tabla proveedores_incidencia (ahí se graban los comprobantes generados)

Cree la vista proveedores_incidencias_ventas. Tomé el caso C1

y le pedí al SP que genere por cantidad 2000. Y en proveedores_incidencias los creó bien:

select SUM(cantidad_final) from PROVEEDORES_INCIDENCIA (solo estan los del C1, que se crearon después de que ejecuté el SP)



[may_Proveedores_articulos_incidencia]

Variable	Ejemplo	Obs
@Cod_articulo	C1	NN - Indica el articulo a generar ventas
@cantidad	200	NN - Indica cantidades a dividir en ventas
@comprobante	20240829	NN- numero de comp (fecha sin guiones)
@fecha_desde	2024-10-6	Null - Por defecto son 6 meses - AGREGAR variable
@fecha_hasta	2024-12-8	Null - Por defecto son 6 meses - AGREGAR variable
@clien_agru	FULL	Null - Indica a que grupo de clientes se debe ejecutar las ventas

Opción 1:

@fecha_desde

@fecha_hasta

Agregar en la vista [full_Proveedores_Incidencia_ventas] el campo comp.FECHA_ALTA real. El campo fecha_desde es desde qué fecha se considera:

```
select
itm.ARTICULO, art.DESCRIP_ARTI, abs(sum(itm.CANT)) as venta_total, comp.CLIENTE, cli.NOM_FANTASIA, cli.AGRU_3
,
Dateadd(Month, Datediff(Month, 0, DATEADD(m, -6, current_timestamp)), 0) as fecha_desde
, comp.FECHA_ALTA /* del comprobante*/
from factu_full_central.dbo.COMP_EMITIDOS as comp
inner join factu_full_central.dbo.RENG_FAC as itm on (comp.TIPO=itm.TIPO_FACT and comp.NUM=itm.NUM_FACT)
inner join factu_full_central.dbo.CLIENTES as cli on comp.CLIENTE=cli.NUM_CLIENTE
inner join factu_full_central.dbo.ARTICULOS as art on itm.ARTICULO=art.COD_ARTICULO
where comp.FECHA >= Dateadd(Month, Datediff(Month, 0, DATEADD(m, -6, current_timestamp)), 0)
and cli.AGRU_3 in ('FULL','PRO')
and comp.ACTUA_STOCK = 'S'
and USA_COMPO = 'N'
and art.SIN_STOCK = 'N'
group by itm.ARTICULO, art.DESCRIP_ARTI, comp.CLIENTE, cli.AGRU_3, cli.NOM_FANTASIA, comp.FECHA_ALTA
```

Con comp.FECHA_ALTA se puede filtrar y la agrupación del cliente ya la tenemos en el campo AGRU_3 corresponde a la tabla clientes

```
use factu_full_central_desa_nacho
select * from full_Proveedores_Incidencia_ventas where ARTICULO = 'C1' AND fecha_alta between '2024-04-01' AND '2024-04-20' and AGRU_3 = 'FULL'
```

En el SP indicar fecha_desde y fecha_hasta si no son null, que asigne los filtros en el inner join a la tabla:

```
-- where a.articulo = @Cod_articulo

@Cod_articulo varchar(10) ,
@cantidad int ,
@comprobante int
as
;

declare @numeracion int =(select isnull(max(nro_ejecucion)+1,1) from proveedores_incidencia);

declare @TEMP2 TABLE (COD_ARTICULO varchar(80),EN_VISTA varchar(80),DESCRIP VARCHAR (330)
,CLIENTE INT
,NOMBRE_FANTASIA VARCHAR(80)
,CANTIDAD NUMERIC (12,4)
)
-----CHECK DE ARTICULOS PARA EL CALCULO-----
insert into @TEMP2(COD_ARTICULO,EN_VISTA,DESCRIP,CANTIDAD,CLIENTE,NOMBRE_FANTASIA)
select distinct COD_ARTICULO,isnull(ARTICULO,'NO') as EN_VISTA,ART.DESCRIPT_ARTI,
CAST(ABS(CHECKSUM(NEWID())) % (25 -1) AS INT) + 1 as cantidad
,CLI_NUM_CLIENTE,CLI_NOM_FANTASIA
-----CHECK DE ARTICULOS PARA EL CALCULO-----
left join full_Proveedores_Incidencia_ventas as piv on art.COD_ARTICULO=piv.ARTICULO
,factu_full_central.dbo.clientes AS CLI
where CLI.RDNO = piv.RDNO
AND ART.COD_ARTICULO = @Cod_articulo

declare @en_vista varchar(10)

select @en_vista = en_vista from @TEMP2

if @Cod_articulo = @en_vista
begin
```

67 % < 67 % Messages

SP - Pedidos

Cargar SP de pedidos -

SP - Transferencia

sp full_transferencia_mayo

El SP se ejecuta únicamente en el Pedido Único

- Transfiere mercadería desde el depósito Único (con partida) hacia el depósito "MAY"

Variable	Tipo	Ejemplo	OBS
@tipo	varchar(3)	STO	Tipo de movimiento en RP
@cod_articulo	varchar(30)	C8	Artículo que se va a transferir entre depósito
@cod_partida	varchar(25)	'2022103'	Nombre de la partida
@depo_ori	varchar(3)	'DEP'	Depósito de origen de partida
@depo_desti	varchar(3)	'MAY'	Depósito a transferir partida de del artículo
@cantida	numeric(18,4)	300	Cantidades a transferir de esa partida
@fecha	datetipe	20241003	Dato de transferencia
@pedi_tipo	varchar(3)	DDW	Va el valor del tipo de pedido que cargan en Mayorista - Es solo para registrar la trazabilidad de la transferencia.
@pedi_num	int	2329	Va el valor del número del pedido que cargan en Mayorista - Es solo para registrar la trazabilidad de la transferencia.

Requerimiento - Desarrollo Mayorista

Pantalla 1

Login - credenciales locales

- Empresa (base de datos)
- Usuario
- Contraseña

*Todo local en el desarrollo, no auth contra la base.

Pantalla 2

Inicio - Dashboard principal

Pedido Deposito Unico

Pedido Mayorista

Ventas Adicionales

Pantalla 3

Pedido Mayorista

- Informacion de Pedido

*campos manuales por el usuario - no se autocompletan

*El campo de tipo de pedido tiene que traer siempre que la columna DEPO = MAY (tabla COMP)

Agregar modal para busqueda de cliente por numero / razon / nombre fantasia

DB - pedi_cabe / clientes

- Items del Pedido

Código	Nombre	Cantidad	Costo	Precio	P. elegidas	
C9 1	Fernet Branca - 750 ml	1 2	\$759.26	\$0.00	4 + 3	Partida Eliminar
		1	\$0.00	\$0.00	+ +	Partida Eliminar
		1	\$0.00	\$0.00	+ +	Partida Eliminar

Agregar modal de busqueda de articulos, por sku o descrip
el campo cantidad no puede superar el stock disponible por partida
P. elegidas = puede visualizar las partidas

DB - pedi_items / articulos / compro_partidasDetalles

- Toma siempre y muestra stock del deposito (Mayorista)
- Costo y partidas del deposito Mayorista

Pantalla 4

Pedido Deposito unico

- *Mismo formulario que Pedido Mayorista - infomación de pedidos e items*

Información del pedido

Número de cliente	Tipo	Vendedor:
7776	Q	Pedido Mayorista
Lista	Razón	Nombre fantasia
<input style="width: 100%; height: 30px; border: 1px solid black; margin-bottom: 5px;" type="text"/> <input style="width: 100%; height: 30px; border: 1px solid black; margin-bottom: 5px;" type="text"/>		
Fecha de entrega		
<input style="width: 100%; height: 30px; border: 1px solid black; margin-bottom: 5px;" type="text"/> 31/08/2023		

Código	Nombre	Cantidad	Costo	Precio	P. elegidas		
C9	Fernet Branca - 750 ml	1	\$759.26	\$0.00	+ Partida	Eliminar	
		1	\$0.00	\$0.00	+ Partida	Eliminar	
		1	\$0.00	\$0.00	+ Partida	Eliminar	

- Toma stock de DEP (Deposito Unico)
- Carga datos del pedido cliente, se genera un pedido (pedi_cabe pedi_items).
- Se transfiere del deposito DEP a Mayorista
- Se crea registro en la reng_fact
- Picking (Deposito)

Pantalla 5

- Ventas Adicionales

Genera, por sp o por formulario, las ventas adicionales en la tabla (ren_fact2)

Pantalla de formulario

SKU	DESCRIP	CANTIDAD	PRECIO	CLIENTE	S/C	DESCUENTO

La idea es colocar un btn swich que lo que permite es habilitar o no la opcion de SP (dinamico) o imptador (directo)

• Por SP

Se importan un template en excel, indicando variables de como va a funcionar el SP.

El SP recibe como variables, clientes, fecha desde, hasta, sku, cantidad, precio y S/C

El usuario puede completar los datos de manera manual (con el numero resuelve el nombre de la razon social , y con el SKU, la descripcion del articulo)

- Cliente

- Puede ser lista de clientes (importado por id_cliente - [file](#))
- o puede seleccionar tipos de clientes (tabla - agrupacion clientes) - [desplegable con agrupacion de clientes](#)

- Fecha desde - hasta

- Manual - Puede setear por calendar o ingresando la fecha manualmente en dos campos entre fecha, el SP generara los registros en la tabla en relacion a este dato.

- Template ([Excel](#))

	A	B	C	D	E
1	Cliente	sku	Precio	Cantidad	S/C
2					
3					
4					

Con estos datos, se completan los requerimientos de SP y una vez completo el formulario,

- [Directo desde el Formulario](#)

Completando el formulario, de manera manual o importada desde Excel, se inserta directo a la tabla de ren_fact2

Pantalla de formulario

SKU	DESCRIP	CANTIDAD	PRECIO	CLIENTE	S/C	DESCUENTO

- Template ([Excel](#))

	A	B	C	D	E
1	Cliente	sku	Precio	Cantidad	S/C
2					
3					
4					

Modificaciones

- Modificaciones 02/9

INICIO

- ✓ Cambiar color de fuente en errores - rojo

Combinación Incorrecta

BUSCADOR

- Validar con enter o seleccionando directamente - Doble click - mejorar posición de boton
 - En cualquiera de los casos, cerrar modal

13581	ana belen quintana beccuti	No Definido
7295	Leonardo Cuccurullo	Leonardo

CONFIRMAR SELECCIÓN

Fecha por defecto (HOY)

dd/mm/aaaa

Si me arrepenti de buscar por ID, ya no me deja cambiar la opción

Buscar Cliente

101

No hay resultados.

CONFIRMAR SELECCIÓN

Manejo de errores en Rojo

No hay resultados.

PANTALLA PRINCIPAL

- ✓ Traer información del cliente cargado --

The screenshot shows a top navigation bar with a search field ('BUSCAR CLIENTE'), a date field ('02/09/2024'), and three buttons: 'CONFIRMAR', 'VER PEDIDO', and 'FINAL'. Below this is a table header with columns: 'Lista' (highlighted with a red box), 'Vendedor', and 'Tipo'. Under 'Lista' are dropdowns for 'zzCentral Bs As' and 'No Definido'. Under 'Tipo' is a dropdown set to 'Pedido Web'. The table body has columns: 'Código', 'Nombre', 'Cantidad', 'Costo Total', 'Precio Total', and 'Opciones'. Each row contains input fields for 'Código' (with a search icon), 'Nombre', 'Cantidad', and 'Costo Total'. The 'Precio Total' column has a '\$' sign and 'Precio Total' text. The 'Opciones' column features a blue button labeled 'PARTIDA' and a red button. A large red box highlights the 'Lista' column header.

- ✓ Traer cantidad disponible de partidas (Suma)

The screenshot shows a form with a 'Cantidad' input field (highlighted with a red box) and a 'Costo' input field below it. The 'Costo' field has a '\$' sign and a dropdown menu showing 'Cantidad'.

- ✓ Descolorar cuando no esta disponible (gris - azul)

The screenshot shows a 'Opciones' section with a blue button labeled 'PARTIDA' and a red button next to it.

- ✓ Cuando buscas el cliente - Focus en Código de articulo

BUSCAR CLIENTE	<input type="text" value="dd/mm/aaaa"/>	
Lista	Vendedor	
Mercado Libre AMBA	No Definido	
Código	Nombre	Cantidad
<input type="text"/> Código	<input type="text"/> Nombre	<input type="text"/> Cantidad
	No hay resultados	

- Agregar búsqueda por nombre

Código	Nombre	Cantidad. Disponibles: 6792	Costo Total
c83	100 Pipper - 750 ml	800	\$ 958.048,00
Código	Nombre	Cantidad	Costo Total
			\$ Costo Total

- ✓ Una vez enviado el pedido, limpiar dash

BUSCAR CLIENTE

10/09/2024

DESCONFIRMAR X

VER PEDIDO *

FINALIZAR ▾

Cliente Seleccionado: 14859 - Diego Amden**Lista**

Lista Ecommerce

Vendedor

Edgard

Tipo

Pedido Web

Código

c83

**Nombre**

100 Pipper - 750 ml

Cantidad Disponibles: 6792**Costo Total**

\$ 958.048,00

Precio Total

\$ 1.552.000,00

Opciones

PARTIDA

**Código**

Nombre

**Nombre**

Cantidad

\$ Costo Total

\$ Precio Total

PARTIDA



AGREGAR FILA

PARTIDAS

- Al "confirmar selección" , se quite la pantalla de "PARTIDAS"

255	720	DEP	1520.81	jj23
-----	-----	-----	---------	------

CONFIRMAR SELECCIÓN

- En el pedido Mayorista

Debe mostrar únicamente las partidas disponibles del deposito "MAY"

Pedido Unico **Pedido Mayoris**

BUSCAR CLIENTE Q 10/09/2024

Cliente Seleccionado: **12675 - DA**

Lista Mercado Libre AMBA

Código c9 **Nombre** Fernet Br

Asignar Partidas Cantidad A Favor: **300** Cantidad Seleccionada: **0** X

Numero	Disponibles	Deposito	Costo Unitario	Ubicación	Definir Cantidad
c90	No Definido	DEP	1520.81	1	<input type="text"/>
nacho_prueba	208	DEP	1520.81	asdasd	<input type="text"/>
c90	No Definido	DEP	1520.81	1	<input type="text"/>
c90	No Definido	DEP	1520.81	1	<input type="text"/>
c90	1	TRA	1520.81	1	<input type="text"/>

CONFIRMAR SELECCIÓN

Modificaciones 06-09

Avisar que numero de pedido creo --

Ver E Car 8 Cantidad \$ Costo Total \$ Precio Total \$

Operación Exitosa

Los datos han sido actualizados correctamente.

- ✓ Al generar el pedido, no carga las ventas adicionales

order by FECHA_ALTA desc

	TIPO	NUM	CLIENTE	FECHA	FECHA_ENTRE	TOTAL	OBS	ESTADO	CODI_VENDE	MONE	MONE_COTI	USER_ALTA	USER_MODI	LIST
1	DDW	10948	1	2024-09-10 00:00:00.000	NULL	NULL	NULL	PEN	EDG	PES	NULL	pediweb	NULL	ECC

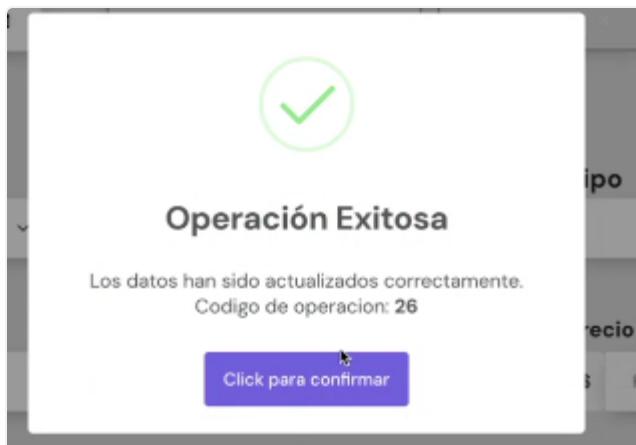
	TIPO	NUM	RENGLON	ARTICULO	CANT	PRECIO	OBS1	OBS2	OBS3	OBS4	CANT_PEND	PORCEN_DESCUEN	CODI_ESTA	BULTOS	ITEM_AC
1	DDW	10948	1	c83	800.0000	1552000.0000	NULL	NULL	NULL	NULL	800.0000	NULL	PRE	0.0000	NULL

	ARTICULO	DESCRIP_ARTI	venta_total	CLIENTE	NOM_FANTASIA	AGRU_3	fecha_desde	FECHA_ALTA
1	C83	100 Pipper - 750 ml	6.0000	1065	Fullescario Nuñez	FULL	2024-03-01 00:00:00.000	2024-06-19 11:16:34.780
2	C83	100 Pipper - 750 ml	6.0000	1014	Fullescario Mendoza	FULL	2024-03-01 00:00:00.000	2024-06-14 08:05:59.280
3	C83	100 Pipper - 750 ml	2.0000	1067	Fullescario San Miguel	FULL	2024-03-01 00:00:00.000	2024-06-07 09:02:13.450
4	C83	100 Pipper - 750 ml	2.0000	1069	Fullescario Florencio Varela	FULL	2024-03-01 00:00:00.000	2024-06-04 10:42:53.510
5	C83	100 Pipper - 750 ml	2.0000	1066	Fullescario Congreso	PRO	2024-03-01 00:00:00.000	2024-06-03 06:55:25.883
6	C83	100 Pipper - 750 ml	4.0000	1023	Fullescario Neuquen	FULL	2024-03-01 00:00:00.000	2024-05-31 13:08:51.100
7	C83	100 Pipper - 750 ml	6.0000	1065	Fullescario Nuñez	FULL	2024-03-01 00:00:00.000	2024-05-30 09:24:53.623
8	C83	100 Pipper - 750 ml	4.0000	1032	Fullescario Wilde	FULL	2024-03-01 00:00:00.000	2024-05-28 11:05:12.860
9	C83	100 Pipper - 750 ml	4.0000	1065	Fullescario Nuñez	FULL	2024-03-01 00:00:00.000	2024-05-22 12:42:23.213
10	C83	100 Pipper - 750 ml	3.0000	1064	Fullescario Recoleta	PRO	2024-03-01 00:00:00.000	2024-05-20 11:50:52.510

Query executed successfully.

Modificaciones 13/09

- ✓ Especificar el tipo de pedido y numero



- ✓ Ver detalle de partidas en nuevo buscador por articulos

Buscar Codigo de Articulo por Descripcion

BUSCAR

SKU	Descripcion
C22	Malibu - 750 ml

- Quitar agrupacion y fecha desde - hasta por SP
- Validar el excel cuando falta un dato en Ventas adicionales - Carga dinamica

Modificaciones - 27-9

- Comunicar tiempos de cargas en busqueda de articulos
- Buscador de artitulos que no funciona bien
- Agregar en Ventas adicionales el fecha desde hasta y categoria
- Agregar tiempos de carga en importador - masivo

Pedido Unico Pedido Mayorista **Ventas Adicionales** LOG OUT

Metodo

Importar Directo ▼ VER PEDIDO FINALIZAR

Subir Excel. Ver Plantila

Seleccionar archivo Directo-Brancamenta.xlsx SUBIR

[Archivo clientes on Branca mента . gin blu 24.09.2024.xlsx](#)

- Manejo de errores en el importador
 - Que indique mas espefisifco cual es el error al importar (entiendo que campos que no tienen que estar vacios, pero cual?)

[Directo-Brancamenta.xlsx](#)

Metodo

Importar Directo

VER PEDIDO

FINALIZAR

Error: ERROR at row 2: Error: Cell can't be empty

Subir Excel. Ver Plantilla

Seleccionar archivo

Ningún archivo seleccionado

SUBIR

- Al ver lo que se importo , no aparece el detalle

[Ver Pedido](#)

X

Power BI

Usuario ☐ reportes_full

Password ☐ reportes_full

KPIs en PBI

KPIs Logistica

```
SELECT cab.FECHA, cab.CLIENTE, COM.DESCRIP , COMP.GRUPO , itm.*  
  
FROM factu_full_central.dbo.PEDI_CABE AS CAB INNER JOIN factu_full_central.dbo.PEDI_ITEMS AS ITM  
ON CAB.TIPO = ITM.TIPO  
AND CAB.NUM = ITM.NUM  
LEFT JOIN factu_full_central.DBO.COMP AS COM  
ON COM.COD_COMP  = ITM.TIPO  
INNER JOIN factu_full_reportes.DBO.full_powerbi_KPI_Logistica_depositos_considerar AS DEP  
ON ITM.DEPO_RETIRO = DEP.DEP0_CODI  
AND DEP.CONSIDERAR = 'S'  
INNER JOIN factu_full_reportes.DBO.full_powerbi_KPI_Logistica_Articulos_considerar AS ART_C  
ON ITM.ARTICULO = ART_C.COD_ARTICULO  
AND ART_C.CONSIDERAR = 'S'  
INNER JOIN factu_full_reportes.dbo.full_powerbi_KPI_Logistica_estados_considerar as EST  
ON ITM.CODI_ESTA = EST.CODI_ESTA  
AND EST.CONSIDERAR = 'S'  
INNER JOIN factu_full_reportes.DBO.full_powerbi_KPI_Logistica_Comprobantes AS COMP  
ON ITM.TIPO = COMP.COD_COMP  
AND COMP.CONSIDERAR = 'S'  
  
WHERE SUBSTRING (CAB.TIPO,1,1) = 'D'  
AND YEAR(CAB.FECHA) = 2023
```

Query

```
use factu_full_central
SELECT
    P.TIPO,
    C.DESCRIP_PEDIDO,
    P.LOTE_NUME,
    P.FECHA_ALTA_FECHA_ALTA_PEDIDO,
    P.LOTE_FECHA_INICIO_PICKING,
    DATEDIFF (minute, P.FECHA_ALTA, P.LOTE_FECHA) PEDIDO_EN_LOTE,
    --Diferencia picking:
    --DATEDIFF (day, P.LOTE_FECHA, max_fecha_armado.FECHA_INICIO_ARMADO) DURACION_DIA_PICKING,
    DATEDIFF (minute, P.LOTE_FECHA, max_fecha_armado.FECHA_INICIO_ARMADO) DURACION_MINUTO_PICKING,
    --P.BULTOS, P.LOTE_CONFIRMADO
    P.FECHA_INICIO_ARMADO,
    P.FECHA_ARMADO,
    --Diferencia packing:
    --DATEDIFF (day, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_DIA_PACKING,
    DATEDIFF (minute, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_MINUTO_PACKING

FROM PEDI_CABE P

INNER JOIN COMP C
ON (P.TIPO = C.COD_COMP)

INNER JOIN factu_full_reportes.dbo.full_powerbi_KPI_Logistica_Comprobantes CL
on (CL.COD_COMP = C.COD_COMP)

INNER JOIN
    (SELECT LOTE_NUME, MIN(FECHA_INICIO_ARMADO) AS FECHA_INICIO_ARMADO
     FROM PEDI_CABE
     GROUP BY LOTE_NUME) AS max_fecha_armado ON P.LOTE_NUME = max_fecha_armado.LOTE_NUME

WHERE YEAR(FECHA_ALTA) >= YEAR(GETDATE())
AND CL.CONSIDERAR = 'S' --calendario
--AND p.LOTE_NUME IN ('26347');
```

Explicación

PICKING

El INICIO_PICKING esta registrado en la tabla PEDI_CABE, en el campo LOTE_FECHA.

Un lote se compone de diferentes pedidos. La alta fecha de alta de cada pedido esta en la tabla PEDI_CABE en el campo FECHA_ALTA.

El fin del lote (FIN_PICKING) esta calculado a partir de la resta entre el INICIO_PICKING y la primer fecha registrada de INICIO_ARMADO (INICIO_PACKING).

- Para calcular cuál es la última fecha registrada de INICIO_ARMADO, se selecciona la primera fecha registrada en FECHA_INICIO_ARMADO en la tabla PEDI_CABE por numero de lote (LOTE_NUME)

```
INNER JOIN
    (SELECT LOTE_NUME, MIN(FECHA_INICIO_ARMADO) AS FECHA_INICIO_ARMADO
     FROM PEDI_CABE
     GROUP BY LOTE_NUME) AS max_fecha_armado ON P.LOTE_NUME = max_fecha_armado.LOTE_NUME
```

- Se saca la diferencia entre el inicio de la fecha y el de inicio del armado, por número de lote, en días y minutos

```
DATEDIFF (day, P.LOTE_FECHA, max_fecha_armado.FECHA_INICIO_ARMADO) DURACION_DIA_PICKING,
```

```
DATEDIFF (minute, P.LOTE_FECHA, max_fecha_armado.FECHA_INICIO_ARMADO) DURACION_MINUTO_PICKING,
```

PACKING

Se saca la diferencia en horas y minutos entre FECHA_INICIO_ARMADO y FECHA_ARMADO de la tabla PEDI_CABE

```
DATEDIFF (day, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_DIA_PACKING,
DATEDIFF (minute, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_MINUTO_PACKING
```

Vista dbo.KPI_logistica_duracion

```
ALTER view [dbo].[KPI_logistica_duracion]
AS
SELECT
    P.TIPO ,
    P.NUM,
    C.DESCRIP,
    P.LOTE_NUME,
    P.FECHA_ALTA FECHA_ALTA_PEDIDO,
    P.LOTE_FECHA INICIO_PICKING,
    DATEDIFF (minute, P.FECHA_ALTA, P.LOTE_FECHA) PEDIDO_EN_LOTE,
    P.LOTE_USUARIO,
    --Diferencia picking:
    --DATEDIFF (day, P.LOTE_FECHA, max_fecha_armado.FECHA_INICIO_ARMADO) DURACION_DIA_PICKING,
    DATEDIFF (minute, P.LOTE_FECHA, CA.FECHA_INICIO_ARMADO) DURACION_MINUTO_PICKING,
    --P.BULTOS, P.LOTE_CONFIRMADO
    P.FECHA_INICIO_ARMADO,
    P.FECHA_ARMADO,
    P.USUARIO_ARMADO,
    --Diferencia packing:
    --DATEDIFF (day, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_DIA_PACKING,
    DATEDIFF (minute, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO) DURACION_MINUTO_PACKING
    --,*  

FROM factu_full_central.dbo.PEDI_CABE P  

    INNER JOIN factu_full_central.dbo.COMP C
        ON (P.TIPO = C.COD_COMP)  

    INNER JOIN factu_full_reportes.dbo.full_powerbi_KPI_Logistica_Comprobantes CL
        on (CL.COD_COMP = C.COD_COMP)  

    INNER JOIN
        (SELECT c.TIPO , c.NUM, c.LOTE_NUME, MIN(c.FECHA_INICIO_ARMADO) AS FECHA_INICIO_ARMADO
         FROM factu_full_central.dbo.PEDI_CABE c
         --where c.lote_num = '26753'
         GROUP BY c.LOTE_NUME, c.NUM, c.TIPO ) AS CA ON P.LOTE_NUME = CA.LOTE_NUME
                                                AND CA.TIPO = P.TIPO
                                                AND CA.NUM = P.NUM  

WHERE YEAR(p.FECHA_ALTA) >= YEAR(GETDATE()) and MONTH(p.FECHA_ALTA) >= MONTH(GETDATE())
    AND CL.CONSIDERAR = 'S' --calendario
    --AND p.LOTE_NUME IN ('26753') and p.num = '124930'--prueba
    --and p.USUARIO_ARMADO is null
    AND p.FECHA_INICIO_ARMADO is not null and fecha_armado is not null
```

```
/*
NOTA 18-04-24
Existen casos donde FECHA_INICIO_ARMADO y FECHA_ARMADO vienen nulls por error de RP. Ya nos
confirmaron que lo solucionaron.
```

Existen casos donde P.LOTE_USUARIO no tiene nada asignado, también se habló con RP y ya no debería suceder.

Por ese motivo, se omiten esos casos y ya no deberían existir.

DURACION_PARCIAL_MINUTO_PACKING es parcial porque se deben hacer un sum, que se hace en la vista KPI_logistica_duracion

```
*/
```

```
GO
```

- Se omiten los casos donde vienen null fecha_inicio_armado y fecha_armado.
 - p.FECHA_INICIO_ARMADO is not null and fecha_armado is not

(Venían null por error de RP)

- DURACION_PARCIAL_MINUTO_PACKING es parcial porque se deben hacer un sum, que se hace en la vista KPI_logistica_duracion.
 - DATEDIFF (minute, P.FECHA_INICIO_ARMADO, P.FECHA_ARMADO)
DURACION_PARCIAL_MINUTO_PACKING
- Para obtener cuándo termina el picking, se consulta la primer fecha_inicio_armado registrada, en el siguiente SELECT:

```
SELECT c.TIPO , c.NUM, c.LOTE_NUME, MIN(c.FECHA_INICIO_ARMADO) AS FECHA_INICIO_ARMADO
      FROM factu_full_central.dbo.PEDI_CABE c
      --where c.lote_num = '26753' --prueba
      GROUP BY c.LOTE_NUME, c.NUM, c.TIPO
```

- Se añade a la consulta principal con un INNER JOIN, considerando el
 - Tipo de pedido
 - Número de pedido
 - Número de lote

```
INNER JOIN
```

```
(SELECT c.TIPO , c.NUM, c.LOTE_NUME, MIN(c.FECHA_INICIO_ARMADO) AS FECHA_INICIO_ARMADO
```

```
FROM factu_full_central.dbo.PEDI_CABE c
--where c.lote_nume = '26753' --prueba
GROUP BY c.LOTE_NUME, c.NUM, c.TIPO ) AS CA
ON P.LOTE_NUME = CA.LOTE_NUME
AND CA.TIPO = P.TIPO
AND CA.NUM = P.NUM
```

Query métrica

```
SELECT
    K.TIPO,
    K.NUM,
    K.DESCRIP,
    K.FECHA_ALTA_PEDIDO,
    K.LOTE_NUME,
    K.PEDIDO_EN_LOTE,
    K.LOTE_USUARIO,
    K.DURACION_MINUTO_PICKING,
    AVG(K.DURACION_MINUTO_PICKING) OVER (PARTITION BY K.LOTE_USUARIO) AS P_PICKING_X_USUARIO, --Promedio
    picking por usuario
    AVG(K.DURACION_MINUTO_PICKING) OVER () PROMEDIO_PICKING, --promedio picking general

    K.USUARIO_ARMADO,
    K.DURACION_MINUTO_PACKING,
    AVG(K.DURACION_MINUTO_PACKING) OVER (PARTITION BY K.USUARIO_ARMADO) AS P_PACKING_X_USUARIO, --
    promedio packing por usuario
    AVG(K.DURACION_MINUTO_PACKING) OVER () PROMEDIO_PACKING, --promedio packing general

    count (*) OVER (PARTITION BY K.TIPO) CANT_PEDI_X_TIPO --cantidad de pedidos por tipo

    FROM factu_full_reportes.[dbo].[KPI_logistica_duracion] K
```

- Promedio de duración de picking por usuario:

```
AVG(K.DURACION_MINUTO_PICKING) OVER (PARTITION BY K.LOTE_USUARIO) AS P_PICKING_X_USUARIO
```

- Promedio de duración de picking general:

```
AVG(K.DURACION_MINUTO_PICKING) OVER () PROMEDIO_PICKING
```

- Promedio de duración de packing por usuario:

```
AVG(K.DURACION_MINUTO_PACKING) OVER (PARTITION BY K.USUARIO_ARMADO) AS P_PACKING_X_USUARIO
```

- Promedio de duración de packing general:

```
AVG(K.DURACION_MINUTO_PACKING) OVER () PROMEDIO_PACKING
```

- Cantidad de pedido por tipo, proceso

```
count (*) OVER (PARTITION BY K.TIPO) CANT_PEDI_X_TIPO
```

PBI - Stocks

Conceptos y definiciones

Sirve para el análisis del stock central, mayorista o el de las sucursales propias, dividiendo el inventario en alta, media y baja rotacion.

Se puede segmentar el inventario por categoría, tipo de producto y proveedor.

El concepto de rotacion se puede ver en Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554>)

Vistas

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11234>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6544>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6584>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11234>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6684>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6644>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6724>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6804>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6664>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11374>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6884>)

Crear la vista rotacion

Tablas

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11494>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11514>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11534>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11454>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11834>)

Versiones del PBI

Actualmente esta en la versión 1.4

Evoluciones

- ~~Sumar una tabla para calcular la rotacion mensual y que esta tabla se grabe todos los meses junto al stock @Hernan Reta~~
 - actualmente lo cargamos a mano en la siguiente tabla pbi_rotacion_historica
- ~~@Augusto generar el Proceso para que todas los primeros dias de los meses se genere la tabla? Ojo que falta el campo YYYYMM en la tabla, tenes que tomar el dia del proceso en formato 2023-01 con el 0 adelante~~
- Sumar un indicador, total venta proyectada vs inventario actual @Hernan Reta
- Sumar el stock valuado con costo de reposicion con IVA @Hernan Reta
- ~~Sumar la tabla Rotacion Histórica al PBI para que adrian no la calcule mas @Hernan Reta~~
- Sumar la rotacion por local
- armar el stock minimo por local
- Armar el sobre stock en funcion a la curva de ventas
- armar el proceso para que el sistema envie pedidos sin tomar en cuenta Kikker

PBI - Stock - Concepto Rotacion

Lógica de rotacion por categoría

- Se toman los últimos 6 meses de ventas por unidades por producto sacando el canal mayorista de las ventas

```
SELECT
    case
        when round(sum(v.cant) / 6,2) > cv.cant/ 6 / w.cant_sku * (1+r.PROCENTAJE_BRECHAS)      then 'Alta'
        when round(sum(v.cant) / 6,2) < cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS) then 'Baja'
        else 'Media' end as Rotacion,
    v.ARTICULO , v.AGRU_1,
    replace(left(convert(varchar,getdate(),23),7),'-','') as YYYYMM,
    round(sum(v.cant),2) as Cat_Total_ventas_Semestral,
    round(sum(v.cant) / 6,2) as Cat_Total_ventas_mensual,
    w.cant_sku as Cat_Cant_SKU,
    cv.cant/ 6 / w.cant_sku as SKU_Total_mensual,
    cv.cant/ 6 / w.cant_sku * (1+r.PROCENTAJE_BRECHAS)   as Alta,
    cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS)   as Baja

    FROM pbi_ventas_locales as v left JOIN pbi_calendario as c
    on v.FECHA = c.YYYYMM
    left join full_powerbi_kpi_empresa_clientes as cl
    on v.CLIENTE = cl.NUM_Cliente
    left join full_powerbi_KPI_empresa_cantidad_sku as w
    on v.AGRU_1 = w.agru_1
    left join pbi_brechas_rotacion r
    on v.AGRU_1 = r.AGRU_1
    left join full_powerbi_KPI_empresa_cantidad_vendida_sku as cv
    on v.AGRU_1 = cv.AGRU_1

    where v.BASE_DATOS  = 'factu_full_central'
    and c.POSICION between 1 and 6
    and cl.CODI_AGRU in ('full', 'prop','MELI','VWE','VINO','PERS')

    group by v.ARTICULO , v.AGRU_1, cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS), cv.cant/ 6 /
w.cant_sku * (1+r.PROCENTAJE_BRECHAS),
    w.cant_sku, cv.cant/ 6 / w.cant_sku, r.PROCENTAJE_BRECHAS
```

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11514>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11634>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11254>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11754>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11794>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11814>)

Consideraciones

Se toman los productos que están dentro como componentes para obtener la rotacion de cada uno.

Tambien se toman los PR

Es un subprocesso de Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984>)

Files de armado y control

[Concepto de rotacion.xlsx](#)

[Logica de Rotacion por Categorias.xlsx](#)

SP que se corren automaticamente – Base de reportes

-> pbi_calendario_update
-> pbi_rotacion_historica_insert

Log de Stock

feb 2024 a las 13:46, <hernan.reta@fullescabio.com> escribió:

Tabla Log Stock

Campo	Tipo	Observaciones
DEPO_CODI	varchar(3)	Deposito en donde ocurre
COD_ARTICULO	varchar(30)	Codigo de articulo
CANT_STOCK	numeric(12, 4)	Cantidad de stock disponible al momento
ACCION	varchar(1)	Dos valores -> C -> Cero stock / S -> Ingreso Stock
FECHA	datetime	Fecha con hora en cuando ocurre

Solo corre para los casos en donde el deposito cumple las siguientes condiciones

```
use factu_full_central
select *
from DEPOSITOS
WHERE INACTIVO = 'N' AND GLN = 'S'
```

SQL

```
select *
from FULL_LOG_STOCK
```

Documentacion Vistas y Tablas

V-full_powerbi_kpi_empresa_articulos

Se usa para obtener todos los productos de la base central con el dato del proveedor siempre y cuando sea el proveedor principal

```
select art.COD_ARTICULO,
art.DESCRIP_ARTI,
art.PRECIO_UNI ,
round(art.PORCEN_IMPUES / 100 * art.PRECIO_UNI,2) as Imp_Interno,
round(art.PRECIO_UNI * 1.21 + (art.PORCEN_IMPUES / 100 * art.PRECIO_UNI ),2) as Precio_c_IVA,
art.USA_COMPO ,
art.ACTIVO,
art.AGRU_1 AS AGRUPACION,
art.AGRU_2 AS MARCA,
ppa.PROVE AS PROVEEDOR,
ART.DIAS_DEMORA_ENTRE AS DEMORA,
isnull(agr.codi_agru,90) as Proceso,
art.um_conver_compra as UM_Compra,
ART.PORCEN_IMPUES AS II

from factu_full_central..ARTICULOS  as ART left join factu_full_central..PROVE_ARTI as PPA
on art.COD_ARTICULO = ppa.ARTICULO
and ppa.PROVE_PRINCIPAL = 's'
left join factu_full_central..agrupaciones as AGR
ON ART.AGRU_3 = AGR.CODI_AGRU
AND AGR.NUM_AGRU = 3
```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

V-full_powerbi_kpi_empresa_clientes

Todos los clientes de la base central siempre y cuando esten activos

```
select NUM_Cliente,
case when isnull(nom_fantasia,'')='' then razon
     else nom_fantasia end as NOM_FANTASIA,
IMPOR_LIMI_CREDI, AGR.DESCRIP_AGRU ,
AGR.CODI_AGRU
from factu_full_central..CLIENTES as cli
inner join factu_full_central..AGRUP_CLIENTES as AGR
on CLI.AGRU_3 = AGR.CODI_AGRU

where --AGRU_3 = 'FULL'
cli.estado = 'A'
```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	

Cambios

- 20-09-2023 - se suma a la tabla el campo codi_agru para usarlo en el sql de rotacion para aplicar filtros desde el codigo [@Hernan Reta](#)

Vista - full_powerbi_kpi_empresa_cobros

----- GRUPO A - FACTURAS COMPLETAMENTE PAGADAS

```
SELECT 'FULL' AS EMPRESA,
CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) AS FACTURA_NRO,
COM.ESTADO,
COM.ESTADO AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA), '-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0', MONTH(CE.FECHA))
     ELSE CONCAT(YEAR(CE.FECHA), '-',MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA,CE.FECHA) AS DIAS,
0 AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
      END AS GAP_COBRANZAS,
'PAGADO' AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) <=DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
'NO' END AS AUDITORIA,
COM.OBSER
```

```

FROM factu_full_central..COMPRO_API CA
    INNER JOIN factu_full_central..COMP C
        ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
        ON CA.TIPO = CE.TIPO
        AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
        ON COM.TIPO = CA.API_TIPO
        AND COM.NUM = CA.API_NUM
    LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 = 'FULL'
AND CONCAT (CA.API_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.API_NUM ),8)) IN
    (SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
    FROM factu_full_central..COMP_EMITIDOS AS COM
    WHERE COM.TIPO LIKE 'F%'
    AND COM.ESTADO = 'CAN'
    AND COM.TOTAL <> 0
    AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())))
)

```

UNION ALL

----- GRUPO B - FACTURAS SIN PAGOS EFECTUADOS

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO,'-TOT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,

```

```

CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'
AND COM.SALDO = COM.TOTAL

```

UNION ALL

----- GRUPO C.0 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO LOS PAGOS

```

SELECT 'FULL' AS EMPRESA,
CONCAT (CA.API_TIP0, '-' ,RIGHT (CONCAT ('0000000000', CA.API_NUM ),8)) AS FACTURA_NRO,
'CAN' AS ESTADO,
CONCAT(COM.ESTADO, '-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(DW, COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK, COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK, COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK, CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))

```

```

        ELSE CONCAT(YEAR(COM.FECHA), '-' ,MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0' ,MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-' ,MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0' ,MONTH(CE.FECHA))
     ELSE CONCAT(YEAR(CE.FECHA), '-' ,MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
0 AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA, CE.FECHA) AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
     ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
END AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
'NO' END AS AUDITORIA,
COM.OBSER

FROM factu_full_central..COMPROMO_API CA
    INNER JOIN factu_full_central..COMP C
    ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
    ON CA.TIPO = CE.TIPO
    AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
    ON COM.TIPO = CA.APLI_TIPO
    AND COM.NUM = CA.APLI_NUM
    LEFT JOIN factu_full_central..CLIENTES AS CLI

```

```

ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 = 'FULL'
AND CONCAT (CA.API_NUM,'-',RIGHT (CONCAT ('0000000000',CA.API_NUM ),8)) IN
(SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
FROM factu_full_central..COMP_EMITIDOS AS COM
WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO = 'PEN'
AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())))

UNION ALL
-----
----- GRUPO C.1 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO EL SALDO DE LAS FACTURAS
-----
SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO AS ESTADO,
CONCAT(COM.ESTADO,'-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO)
ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 7 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 15 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 21 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'

```

```

WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO IN ('PEN')
AND COM.SALDO <> 0
AND COM.SALDO <> COM.TOTAL
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'

```

```
UNION ALL
```

```
----- GRUPO D - ANTICIPOS SIN APLICAR
```

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO,'-ANT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
     ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,

```

```

'SIN VENCER' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28  THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'A%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'
AND COM.SALDO = COM.TOTAL

```

Reporte que usa la vista	Link

Vista -

full_powerbi_kpi_empresa_cobros_mayoristas

----- GRUPO A - FACTURAS COMPLETAMENTE PAGADAS -----

```
SELECT 'FULL' AS EMPRESA,
CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) AS FACTURA_NRO,
COM.ESTADO,
COM.ESTADO AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA), '-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0', MONTH(CE.FECHA))
     ELSE CONCAT(YEAR(CE.FECHA), '-',MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA, CE.FECHA) AS DIAS,
0 AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
     WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
     WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
     WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
     WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
     WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
     END AS GAP_COBRANZAS,
'PAGADO' AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <=DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
```

```

' NO' END AS AUDITORIA,
COM.OBSER

FROM factu_full_central..COMPROMISOS CA
    INNER JOIN factu_full_central..COMPRES C
        ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
        ON CA.TIPO = CE.TIPO
        AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
        ON COM.TIPO = CA.APLI_TIPO
        AND COM.NUM = CA.APLI_NUM
    LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 in ( 'MAYOR', 'VINO', 'VEN', 'MULET', 'QUILM' )
AND CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) IN
    (SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
    FROM factu_full_central..COMP_EMITIDOS AS COM
    WHERE COM.TIPO LIKE 'F%'
    AND COM.ESTADO = 'CAN'
    AND COM.TOTAL <> 0
    AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())) )

```

UNION ALL

----- GRUPO B - FACTURAS SIN PAGOS EFECTUADOS

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO,'-TOT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS APLI_IMPOR,

```

```

COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
      ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate()))
AND CLI.AGRU_3 in ( 'MAYOR', 'VINO', 'VEN', 'MULET', 'QUILM' )
AND COM.SALDO = COM.TOTAL

```

```

UNION ALL
----- GRUPO C.0 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO LOS PAGOS
-----
```

```

SELECT 'FULL' AS EMPRESA,
CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) AS FACTURA_NRO,
'CAN' AS ESTADO,
CONCAT(COM.ESTADO,'-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(dw, COM.FECHA ) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK, COM.FECHA ) AS SEMANA_EMISION,
DATEPART (WEEK, COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,

```

```

DATEPART (WEEK, CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
      ELSE CONCAT(YEAR(COM.FECHA), '-' ,MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
      ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-' ,MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0', MONTH(CE.FECHA))
      ELSE CONCAT(YEAR(CE.FECHA), '-' ,MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
0 AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA, CE.FECHA) AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
      ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
      END AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
      END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
'NO' END AS AUDITORIA,
COM.OBSER

FROM factu_full_central..COMPROMO_API CA
    INNER JOIN factu_full_central..COMP C
        ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
        ON CA.TIPO = CE.TIPO
        AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
        ON COM.TIPO = CA.APLI_TIPO

```

```

        AND COM.NUM = CA.APLI_NUM
        LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 in ( 'MAYOR', 'VINO', 'VEN', 'MULET', 'QUILM' )
AND CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) IN
    (SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
    FROM factu_full_central..COMP_EMITIDOS AS COM
    WHERE COM.TIPO LIKE 'F%'
    AND COM.ESTADO = 'PEN'
    AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())))
UNION ALL
-----
----- GRUPO C.1 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO EL SALDO DE LAS FACTURAS
-----
SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO AS ESTADO,
CONCAT(COM.ESTADO,'-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
      ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
      ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
      ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS APLI_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO)
      ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 7 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 15 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'

```

```

WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO IN ('PEN')
AND COM.SALDO <> 0
AND COM.SALDO <> COM.TOTAL
AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate()))
AND CLI.AGRU_3 in ('MAYOR', 'VINO', 'VEN', 'MULET', 'QUILM')

```

```
UNION ALL
```

```
----- GRUPO D - ANTICIPOS SIN APLICAR
```

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO, '-', RIGHT (CONCAT ('0000000000', COM.NUM), 8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO, '-ANT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW, COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK, COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK, COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK, COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
ELSE CONCAT(YEAR(COM.FECHA), '-', MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-', MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-', MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
ELSE 0 END AS DIAS_AL_VENCIMIENTO,

```

```

COM.CLIENTE,
CLI.NOM_FANTASIA,
'SIN VENCER' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'A%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 in ( 'MAYOR', 'VINO', 'VEN', 'MULET', 'QUILM' )
AND COM.SALDO = COM.TOTAL

```

Reporte que usa la vista	Link

V-full_powerbi_kpi_empresa_cuentas_resultado

```
select cuenta, DESCRIP CUENTA,  
  
CASE WHEN substring(cuenta,1,5) = 41300 THEN 'VENTAS SUC PROPIAS'  
WHEN substring(cuenta,1,5) = 41113 THEN 'VENTAS ECOMMERCE'  
WHEN substring(cuenta,1,5) = 41126 THEN 'VENTAS ECOMMERCE'  
  
ELSE 'VENTAS CANAL' END AS CONSOLIDACION,  
CONCAT(CUENTA,' II') AS IMPUESTOS INTERNOS  
  
from factu_full_central..PLAN_CUENTAS  
where CUENTA in (  
'41111',  
'41300FL',  
'41300BE',  
'41300PA',  
'41300CE',  
'41300IX',  
'41300MS',  
'41300RC',  
'41300VIII',  
'41300X',  
'41113',  
'41113FU',  
'41126',  
'41114',  
'41114VE',  
'41120',  
'41118',  
'41300V',  
'41300MK')  
  
union all  
  
select cuenta, DESCRIP CUENTA,  
CASE WHEN substring(cuenta,1,5) = 41300 THEN 'VENTAS SUC PROPIAS'  
WHEN substring(cuenta,1,5) = 41113 THEN 'VENTAS ECOMMERCE'  
WHEN substring(cuenta,1,5) = 41126 THEN 'VENTAS ECOMMERCE'  
  
ELSE 'VENTAS CANAL' END AS CONSOLIDACION,  
CONCAT(CUENTA,' II') AS IMPUESTOS INTERNOS
```

```

from factu_full_dist_central..PLAN CUENTAS
where CUENTA in (
'41111',
'41300VC',
'41300FL',
'41300BE',
'41300PA',
'41300CE',
'41300IX',
'41300MS',
'41300RC',
'41300VIII',
'41300X',
'41113',
'41113FU',
'41126',
'41114',
'41114VE',
'41120',
'41118',
'41300V',
'41300MK') ) as T
GO

```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

Cuando se agrega una nueva sucursal es necesario sumar la cuenta de venta de RP central para que se refleje dentro de la busqueda

V-full_powerbi_kpi_empresa_depositos

```
select DEPO_CODI,DESCRIP, 'SI' as KPI_Stock, 'NO' as Pickup
from factu_full_CENTRAL.dbo.depositos

UNION ALL
-- PRIMERA PARTE ES PARA LOS DEPOSITOS QUE ESTAN EN LA BASE DE LOCALES QUE SE SON PROPIOS
select DEPO_CODI,DESCRIP, 'SI' as KPI_Stock, 'NO' as Pickup
from factu_full_locales.dbo.depositos
WHERE (DEPO_CODI IN ('MAL','PAL','BEL','FLS','CER') OR
      SUBSTRING(DEPO_CODI,1,2) = 'FR' )
      AND DESCRIP NOT LIKE 'FRANKFULL%'

UNION ALL
select DEPO_CODI,DESCRIP, 'SI' as KPI_Stock, 'SI' as Pickup
from factu_full_locales.dbo.depositos
WHERE (DEPO_CODI IN ('MAL','PAL','BEL','FLS','CER') OR
      SUBSTRING(DEPO_CODI,1,2) = 'FR' )
      AND DESCRIP LIKE 'FRANKFULL%'

UNION ALL

select DEPO_CODI,DESCRIP, 'NO' as KPI_Stock, 'NO' as Pickup
from factu_full_locales.dbo.depositos
WHERE (DEPO_CODI NOT IN ('MAL','PAL','BEL','FLS','CER') OR
      SUBSTRING( DEPO_CODI,1,2) <> 'FR' )
      AND DESCRIP NOT LIKE 'FRANKFULL%'

UNION ALL

select DEPO_CODI COLLATE Modern_Spanish_CI_AS,DESCRIP COLLATE Modern_Spanish_CI_AS, 'NO' as KPI_Stock,
      'NO' as Pickup
      from factu_full_bahia_blanca.dbo.depositos
```

Reporte de los depósitos cargados en las diferentes bases, depende del tipo de deposito calcula los campos de KPI_STOCK y Pickup

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

Es necesario poner los nuevos depósitos de las sucursales propias ('MAL','PAL','BEL','FLS','CER','LHE','PIL','CON','MST')

SQL de Depositos

```
select DEPO_CODI,DESCRIP  
from factu_full_locales.dbo.depositos
```

Vista -

full_powerbi_kpi_empresa_ingresos_prom_pos

```
select app.sucursal as NUM_Cliente, CLI.NOM_FANTASIA, SUM(APP.TOTAL_NETO) / 4 as INGRESO_PROM

from factu_full_central..full_apps AS APP left join
    factu_full_central..CLIENTES AS CLI
    ON APP.sucursal = CLI.NUM_CLIENTE
WHERE DATEPART (WEEK,APP.fecha_anticipo) IN (DATEPART (WEEK, GETDATE()),DATEPART
(WEEK,GETDATE())-1,DATEPART (WEEK,GETDATE())-2,DATEPART (WEEK,GETDATE())-3)
AND DATEPART (YEAR,APP.fecha_anticipo ) = DATEPART (YEAR, GETDATE())

GROUP BY app.sucursal, CLI.NOM_FANTASIA
```

Calcula el promedio de total neto por local en las ultimas 3 semanas de la tabla full_apps en la base de central

Reporte que usa la vista	Link

Consideraciones

Vista - full_powerbi_kpi_empresa_presupuesto

```
SELECT  
CASE WHEN LEN(MONTH(PRE.FECHA)) = 1 THEN CONCAT(YEAR(PRE.FECHA), '-0', MONTH(PRE.FECHA))  
      ELSE CONCAT(YEAR(PRE.FECHA), ' - ', MONTH(PRE.FECHA)) END AS YYYYMM,  
PRE.CUENTA AS CUENTA,  
PRE.IMPOR AS MONTO_PRESUPUESTO,  
PRE.FORMALIDAD AS FORMALIDAD  
  
FROM factu_full_central..PRESUP_CONTA as PRE
```

Consulta la tabla de presupuesto contable de la base de Central

Reporte que usa la vista	Link

Consideraciones

Vista

- full_powerbi_kpi_empresa_prov_articulo

Genera un listado de los proveedores dados de alta en la base de central

```
SELECT NUM_PROV, RAZON, NOM_FANTASIA  
FROM factu_full_central.dbo.PROVEEDORES
```

Reporte que usa la vista	Link

Consideraciones

V-full_powerbi_kpi_empresa_stock

```
select
c.COD_ARTICULO,
c.COD_DEPO ,
c.COD_PARTIDA ,
c.CANT_PEND as QTY,
c.COSTO_UNI + c.IMPUES_INTERNOS_UNI COSTO_UNI

from factu_full_central..compro_partidas c
where
c.CANT_PEND > 0
and substring(c.cod_articulo,1,1) in ('C')

union all

SELECT
dep.COD_ARTICULO ,
DEP.DEPO,
'LOCAL' as COD_PARTIDA,
DEP.CANT_STOCK as QTY,
0 as COSTO_UNI

FROM  factu_full_locales..ARTICULOS_STOCK_DEPO AS DEP

where substring(dep.cod_articulo,1,1) in ('C')
```

Toma el stock de todos los artículos C en la base de central y Locales, desde la base de central calcula el stock por los compro partidas, asi puede sacar el impuesto interno por partida y articulo. En la base de locales consulta la tabla del stock de deposito

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	Link

Consideraciones

Vista - full_powerbi_kpi_empresa_ventas_det

```
SELECT
ITM.TIPO_FACT,
TIPO_COMP.DESCRIP,
CONCAT (ITM.TIPO_FACT,'-',RIGHT (CONCAT ('0000000000', ITM.NUM_FACT),8)) AS FACTURA_NRO,
convert(date, COM.FECHA,13) AS FECHA,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA))
END AS YYYYMM,
COM.CUIT_COMPRO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CLI.AGRU_3,
ITM.ARTICULO,
ART.DESCRIP_ARTI,
ITM.CANT * -1 AS CANT,
ITM.PRECIO,
CASE WHEN TIPO_COMP.COD_GRUPO = 'FIS'
      THEN ITM.CANT * PRECIO_NETO * -1.21
      ELSE ITM.CANT * PRECIO_NETO * -1.21 END AS FACTURADO_TOTAL_C_IVA,
CASE WHEN TIPO_COMP.COD_GRUPO = 'FIS'
      THEN ITM.CANT * PRECIO_NETO * -1
      ELSE ITM.CANT * PRECIO_NETO * -1 END AS FACTURADO_TOTAL_S_IVA,
CASE WHEN TIPO_COMP.COD_GRUPO = 'FIS'
      THEN ITM.CANT * ITM.COSTO_UNI * 1.21
      ELSE ITM.CANT * ITM.COSTO_UNI * 1.21 END AS CMV_C_IVA,
CASE WHEN TIPO_COMP.COD_GRUPO = 'FIS'
      THEN ITM.CANT * ITM.COSTO_UNI
      ELSE ITM.CANT * ITM.COSTO_UNI END AS CMV_S_IVA,
ITM.PORCEN_IMPUES_INTER_HISTO AS II_PORC,
ITM.IMPOR_IMPUES_TOTAL_VTA AS II,
COM.ESTADO,
AGR.DESCRIP_AGRU AS GRUPO,
SUBSTRING ( ITM.ARTICULO,1,1) AS TIPO_PRODUCTO,
COM.ASIEN_NUME_CMV,
COM.ASIEN_NUME,
TIPO_COMP.COD_GRUPO,
ITM.COLOR AS PARTIDA,
ART_PROV.PROVE AS COD_PROVEEDOR,
PROV.RAZON AS NOMBRE_PROVEEDOR,
CONVERT(date,DATEADD(DD,-1,DATEADD(MM,0,GETDATE()))),3) AS FECHA_Reporte,
com.REMI_NUME

FROM factu_full_CENTRAL.dbo.RENG_FAC AS ITM LEFT JOIN factu_full_CENTRAL.dbo.COMP_EMITIDOS AS COM
```

```

ON ITM.NUM_FACT = COM.NUM AND ITM.TIPO_FACT = COM.TIPO
LEFT JOIN factu_full_CENTRAL.dbo.ARTICULOS AS ART
ON ITM.ARTICULO = ART.COD_ARTICULO
LEFT JOIN factu_full_CENTRAL.dbo.CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE
LEFT JOIN factu_full_CENTRAL.dbo.AGRUPACIONES AS AGR
ON ART.AGRU_1= AGR.CODI_AGRU
AND AGR.NUM_AGRU= 1
LEFT JOIN factu_full_CENTRAL.dbo.COMP AS TIPO_COMP
ON ITM.TIPO_FACT = TIPO_COMP.COD_COMP

LEFT JOIN factu_full_CENTRAL.dbo.PROVE_ARTI AS ART_PROV
ON ART.COD_ARTICULO = ART_PROV.ARTICULO
AND PROVE_PRINCIPAL = 'S'

LEFT JOIN factu_full_CENTRAL.dbo.PROVEEDORES AS PROV
ON ART_PROV.PROVE = PROV.NUM_PROV

WHERE YEAR(COM.FECHA) IN ('2023')
AND MONTH (COM.FECHA) IN ('1','2','3','4','5','6','7','8','9','10','11','12')
AND (ITM.TIPO_FACT LIKE ('F%') OR
    ITM.TIPO_FACT LIKE ('C%') OR
    ITM.TIPO_FACT LIKE ('B%') )

--AND ART.COD_ARTICULO = 'C9189'
--AND ITM.NUM_FACT = '52639'
--AND ART.COD_ARTICULO LIKE 'PR%'
AND ITM.COMPO_ITEM IS NULL

```

Reporte de Ventas (FC, NC, ND) de la base de central filtrado por todo el año 2023 y por artículos que tengan un proveedor principal activado. No se toman los artículos con componentes dentro del reporte

Reporte que usa la vista	Link

Consideraciones

V-full_powerbi_KPI_Sucursales_Categorias

```
SELECT CODI_AGRU as CODI_AGRU ,  
DESCRIP_AGRU AS DESCRIP_AGRU  
  
FROM factu_full_central..AGRUPACIONES  
WHERE NUM_AGRU = 1  
  
union  
  
SELECT CODI_AGRU as CODI_AGRU ,  
DESCRIP_AGRU AS DESCRIP_AGRU  
  
FROM factu_full_dist_central..AGRUPACIONES  
WHERE NUM_AGRU = 1
```

Genera un listado de categorías aplicables para los artículos en la base de Central y DCB

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

V-full_powerbi_KPI_Sucursales_Componentes

```
SELECT 'COD_ARTICULO;COD_ARTI_COMPO;CANTI;DEPO' AS campo_expor  
  
UNION ALL  
  
SELECT  
    CONVERT(VARCHAR, COD_ARTICULO)+';'+  
    CONVERT(VARCHAR, COD_ARTI_COMPO)+';'+  
    CONVERT(VARCHAR, CANTI)+';'+  
    ISNULL(CONVERT(varchar, DEPO), '') AS campo_expor  
FROM factu_full_locales.dbo.ARTICULOS_COMPO
```

Listado de los artículos que tienen componentes, muestra el sku original junto con el sku de los componentes y cantidades. Solo los artículos de la base de locales

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

V-full_powerbi_KPI_Sucursales_Sucursales

```
SELECT distinct CONVERT(VARCHAR,DEP.DEPO_CODI) as Deposito,
isnull(CONVERT(VARCHAR,CFG.USUARIO ),'')as idSucursal,
isnull (CONVERT(VARCHAR,EMP.AGRUPACION_SUCURSAL),'' ) as GrupoSucursal,
isnull(CONVERT(VARCHAR,DEP.DESCRIP),'' ) as Nombre_Sucursal

FROM factu_full_locales.dbo.DEPOSITOS as DEP left join factu_full_locales.dbo.EMPRESAS as EMP
on dep.DEPO_CODI = emp.COMP_GRUPO
leFT join factu_full_locales.dbo.FULL_ESCABIO_LOCALES_CONFIG as CFG
on dep.depo_codi = cfg.COD_DEPO
where CFG.USUARIO is not null
and dep.DEPO_CODI not in ('FR1','FR2','FR3','FR4','FR5','FR6','FR7','FR8','FR9','F10','F11' --ACA SE
TIENEN QUE PONER LOS CODIGOS DE DEPOSITO DE PICKUP PARA QUE NO LOS TOME EN EL KPI DE SUCURSALES
,'ADR','PGR','CIP','CAB','MH','TLD','Mir','OLI','PAX','PIN','PMA','PRJ','QUI','RME','REC','ST','NAO','SMA
','TDL','GES') --ELIMINAR SUCURSALES SIN MOVIMIENTOS PEDIDO ALEXIS
```

Desde esta vista se consultan los depósitos de la base de locales y junto con la tabla de configuración. Así podemos tomar todos los locales para agregarlos al reporte general. Se agrega un listado donde podemos excluir locales viejos y los depósitos de pickup. Los de pickup se tienen que agregar para no duplique los movimientos en los KPI.

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

V-full_powerbi_KPI_Sucursales_Subcategorias

```
select CODI_AGRU as idSubcategoria, DESCRIP_AGRU as Subcategoria  
from factu_full_locales.dbo.agrupaciones  
where num_agru = '2'
```

Trae los códigos de marca para adjuntarlos a la base de locales

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Vista -

full_powerbi_KPI_Sucursales_Proveedores

```
SELECT 'ARTICULO;PROVE;RAZON' campo_expor  
  
union all  
  
SELECT  
    P.ARTICULO+',' +  
    CONVERT (varchar, P.PROVE) +',' +  
    PRV.RAZON as campo_expor  
FROM factu_full_central.dbo.PROVE_ARTI AS P LEFT JOIN factu_full_central.dbo.PROVEEDORES AS PRV  
ON P.PROVE = PRV.NUM_PROV  
WHERE P.PROVE_PRINCIPAL = 'S'
```

Genera un listado de los artículos que ofrecen cada uno de los proveedores, dentro de este listado solo se toman los artículos que marcaron un proveedor como Principal.

Vista - full_powerbi_KPI_Sucursales_Productos

```
SELECT 'COD_ARTICULO;DESCRIP_ARTI;GRUPO;MARCA_ID;CODIGO_BARRA_ART;DEPO;USUARIO;COD_ARTICULO_GENERAL' AS campo_expor

UNION ALL

SELECT
    CONVERT(VARCHAR, ART.COD_ARTICULO) +';'+
    ISNULL(ART.DESCRIP_ARTI,'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.AGRU_1),'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.AGRU_2),'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.COD_BARRAS),'')+';'+
    ISNULL(ISNULL(ART.DEPO_DEFAULT,U.DEPO_DEFAULT),'')+';'+
    ISNULL(ART.USER_ALTA,'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.CODIGO_PRD),'') AS campo_expor

FROM factu_full_locales.dbo.ARTICULOS AS ART
LEFT JOIN factu_full_locales.dbo.AGRUPACIONES AS GR1
    ON GR1.NUM_AGRU = 1
    AND ART.AGRU_1 = GR1.CODI_AGRU
LEFT JOIN factu_full_locales.dbo.AGRUPACIONES AS GR2
    ON GR2.NUM_AGRU = 2
    AND ART.AGRU_2 = GR2.CODI_AGRU
LEFT JOIN factu_full_locales.dbo.CHR_US AS U
    ON ART.USER_ALTA = usuario_admin_principal
    AND U.DEPO_DEFAULT <> ''

WHERE
--ART.ACTIVO = 'S'
ART.COD_ARTICULO LIKE 'LC%'

UNION ALL

SELECT
    ART.COD_ARTICULO+';'+
    ART.DESCRIP_ARTI+';'+
    ISNULL(CONVERT(VARCHAR, ART.AGRU_1),'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.AGRU_2),'')+';'+
    ISNULL(CONVERT(VARCHAR, ART.COD_BARRAS),'')+';'+
    'FULL;'+
    'FULLESCABIO;'+
```

```

ISNULL(CONVERT(VARCHAR,ART.CODIGO_PRD),'' ) AS campo_expor

FROM factu_full_locales.dbo.ARTICULOS AS ART
LEFT JOIN factu_full_locales.dbo.AGRUPACIONES AS GR1
ON GR1.NUM_AGRU = 1
AND ART.AGRU_1 = GR1.CODI_AGRU
LEFT JOIN factu_full_locales.dbo.AGRUPACIONES AS GR2
ON GR2.NUM_AGRU = 2
AND ART.AGRU_2 = GR2.CODI_AGRU

WHERE
--ART.ACTIVO = 'S'
ART.COD_ARTICULO NOT LIKE 'LC%'

-----PRODUCTOS BAHIA BLANCA-----
UNION ALL

SELECT
ART.COD_ARTICULO COLLATE Modern_Spanish_CI_AS +' ; '+
ART.DESCRIP_ARTI COLLATE Modern_Spanish_CI_AS+' ; '+
ISNULL(CONVERT(VARCHAR,ART.AGRU_1),'' ) +' ; '+
ISNULL(CONVERT(VARCHAR,ART.AGRU_2),'' ) +' ; '+
ISNULL(CONVERT(VARCHAR,ART.COD_BARRAS),'' ) +' ; '+
'FULL;' +
'FULLESCABIO;' +
ISNULL(CONVERT(VARCHAR,ART.CODIGO_PRD),'' ) AS campo_expor

FROM factu_full_bahia_blanca.dbo.ARTICULOS AS ART
LEFT JOIN factu_full_bahia_blanca.dbo.AGRUPACIONES AS GR1
ON GR1.NUM_AGRU = 1
AND ART.AGRU_1 = GR1.CODI_AGRU
LEFT JOIN factu_full_bahia_blanca.dbo.AGRUPACIONES AS GR2
ON GR2.NUM_AGRU = 2
AND ART.AGRU_2 = GR2.CODI_AGRU

WHERE
--ART.ACTIVO = 'S'
ART.COD_ARTICULO LIKE 'LB%'

```

Genera un listado de los artículos de las bases de Locales y Bahia Blanca, desde la base de locales se traen todos los productos articulos y de bahia blanca solo los LB

V-full_powerbi_KPI_Sucursales_Marcas

```
SELECT
    CODI_AGRU as CODI_AGRU,
    DESCRIP_AGRU as DESCRIP_AGRU

FROM factu_full_locales.dbo.AGRUPACIONES

WHERE NUM_AGRU = 2
```

Listado de todas las marcas agregadas en la base de locales

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

V-full_powerbi_KPI_Sucursales_Comprobantes

```
SELECT REPLACE(COD_COMP,'-','X') AS COD_COMP , --SE CAMBIO EL - POR X PARA MEJORAR LOS REPORTES DE PBI
TIPO,
DESCRIP,
PREFIJO,
'Locales' as DB
FROM factu_full_locales..COMP
where tipo in ('A','B','C','M')

union all

SELECT
COD_COMP collate database_default,
TIPO collate database_default,
DESCRIP collate database_default,
PREFIJO collate database_default,
'Bahia Blanca' collate database_default
FROM factu_full_bahia_blanca..COMP
where tipo collate database_default in ('A','B','C','M')

union all

SELECT COD_COMP,
TIPO,
DESCRIP,
PREFIJO,
'Central' as DB
FROM factu_full_central..COMP
where tipo in ('A','B','C','M')
union all

SELECT COD_COMP,
TIPO,
DESCRIP,
PREFIJO,
'DCB' as DB
FROM factu_full_dist_central..COMP
where tipo in ('A','B','C','M')
and cod_comp NOT IN ( 'FB4','FA4','FA6')
```

Listado de comprobantes de todas las bases (Central, Locales, DCB) con los siguientes cambios

- Se remplazaron los caracteres - por X para poder hacer el parceo de las facturas usando el -

- Se limitan comprobantes de DCB porque están duplicados en otras bases, al estar duplicado el PBI calcula esos comprobantes por 2

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Tabla - pbi_cuentas

Se usa para agrupar y ordenar las cuentas de la contabilidad del central

```
[ID_GRUPO] [int] NULL,  
[ID_SUBGRUPO] [int] NULL,  
[CUENTA] [varchar](10) NULL,  
[DESCRIPCION] [varchar](70) NULL
```

Reporte que usa la vista	Link

Consideraciones

Tabla - pbi_cuentas_grupo

Se usa para agrupar y ordenar las cuentas de la contabilidad del central

```
[ID_GRUPO] [int] NULL,  
[GRUPO] [varchar](50) NULL
```

Reporte que usa la vista	Link

Consideraciones

Tabla - pbi_DisplayOrder

Se usa para agrupar y ordenar las cuentas de la contabilidad del central

```
[ID_GRUPO] [int] NULL,  
[GRUPO] [varchar](50) NULL
```

Reporte que usa la vista	Link

Consideraciones

Cuando se agregan sucursales propias nuevas, es necesario modificar la tabla o agregar cuentas y grupos.

Abajo estan los dos sqls, se deben ejecutar en orden.

SQL 1

```
update pbi_DisplayOrder set cuentaDescripcion = 'Sucursal Recoleta' where CuentaDescripcion = 'Sucursal  
Vii'
```

SQL 2

```
update pbi_DisplayOrder set DisplayOrder = 15 where CuentaDescripcion = 'Sucursal Recoleta'
```

Vista - PNL

```
-- FACTURAS DE VENTAS

SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
     ELSE CONCAT(YEAR(CAB.FECHA), '-' ,MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP_CUENTA,
LEFT(PC.DESCRIP_CUENTA,4) AS DEScrip,
PC.TIPO_CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
      WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 1' CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COMP.TIPO,'-',RIGHT (CONCAT ('0000000000',COMP.NUM),8)) AS FACTURA_NRO,
CONVERT(DATE,COMP.FECHA,103) AS FECHA_FACTURA,
COMP.PROVE AS CODIGO,
PRO.RAZON AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 as AGRUPACION,
'FRK' AS EMPRESA

FROM factu_full_central..ASIEN_LINEA AS LIN
LEFT JOIN factu_full_central..PLAN CUENTAS AS PC
ON LIN.CUENTA = PC.CUENTA
LEFT JOIN factu_full_central..ASIEN_CABE AS CAB
ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
LEFT JOIN factu_full_central..COMPRO_CABE_PROVE AS COMP
ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
AND COMP.ESTADO <> 'ANU'
LEFT JOIN factu_full_central..PROVEEDORES AS PRO
ON COMP.PROVE = PRO.NUM_PROV
LEFT JOIN factu_full_central..COMP AS TCOMP
ON COMP.TIPO = TCOMP.COD_COMP
```

```

LEFT JOIN factu_full_central..COMP_EMITIDOS AS COM
ON LIN.ASIEN_NUME = COM.ASIEN_NUME
AND COM.ESTADO != 'ANU'
LEFT JOIN factu_full_central..CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE YEAR(CAB.FECHA) IN (2022,2023)

AND CAB.ASIEN_TIPO IN ('PRO','PAG','PRI','PAI','GEN','COB','PRI','PAI')
-----  

AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS

UNION ALL
-----  

-- ASIENTOS DE CMV
-----  

SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
ELSE CONCAT(YEAR(CAB.FECHA), '-', MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP_CUENTA,
LEFT(PC.DESCRIP_CUENTA,4) AS DESCRIP,
PC.TIPO_CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 2' CONTROL_PRESUP,
CAB.ASIEN_TIPO,
LIN.COMPROBANTE AS FACTURA_NRO,
--COM.NUM AS FACTURA_NRO,
CONVERT(DATE,COMP.FECHA,103) AS FECHA_FACTURA,
LIN.NUM_CLIENTE AS CODIGO, -- ACA PONGO EN CODIGO EL CAMPO OBSERVACIONE DEL VOUCHER PARA OBTENER EL NRO
DE CLIENTE
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 AS AGRUPACION,
'FRK' AS EMPRESA

```

```

FROM factu_full_central..ASIEN_LINEA AS LIN
    LEFT JOIN factu_full_central..PLAN_CUENTAS AS PC
        ON LIN.CUENTA = PC.CUENTA
    LEFT JOIN factu_full_central..ASIEN_CABE AS CAB
        ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
    LEFT JOIN factu_full_central..COMPRO_CABE_PROVE AS COMP
        ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
        AND COMP.ESTADO <> 'ANU'
    LEFT JOIN factu_full_central..PROVEEDORES AS PRO
        ON COMP.PROVE = PRO.NUM_PROV
    LEFT JOIN factu_full_central..COMP AS TCOMP
        ON COMP.TIPO = TCOMP.COD_COMP
    LEFT JOIN factu_full_central..COMP_EMITIDOS AS COM
        ON ( LIN.ASIEN_NUME = COM.ASIEN_NUME and COM.ESTADO != 'ANU' )
    LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON LIN.NUM_CLIENTE = CLI.NUM_CLIENTE

```

WHERE YEAR(CAB.FECHA) IN (2022,2023)

AND CAB.ASIEN_TIPO IN ('CMV')

AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
 '51216','51219','51214','51256','51257','51251','51226','51227',
 '51228','51229','21116','11112','11111','11132','11133','11131',
 '11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS

UNION ALL

-- PROVEEDORES

```

SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
    ELSE CONCAT(YEAR(CAB.FECHA), '-', MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP_CUENTA,
LEFT(PC.DESCRIP_CUENTA,4) AS DESCRIP,
PC.TIPO_CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR

```

```

WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 3' CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FC_FACTURA_NRO,
CONVERT(DATE,COM.FECHA,103) AS FECHA_FACTURA,
COM.CLIENTE AS CODIGO,
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 as AGRUPACION,
'FRK' AS EMPRESA

```

```

FROM factu_full_central..ASIEN_LINEA AS LIN
LEFT JOIN factu_full_central..PLAN CUENTAS AS PC
ON LIN.CUENTA = PC.CUENTA
LEFT JOIN factu_full_central..ASIEN_CABE AS CAB
ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
LEFT JOIN factu_full_central..COMPRO_CABE_PROVE AS COMP
ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
AND COMP.ESTADO <> 'ANU'
LEFT JOIN factu_full_central..PROVEEDORES AS PRO
ON COMP.PROVE = PRO.NUM_PROV
LEFT JOIN factu_full_central..COMP AS TCOMP
ON COMP.TIPO = TCOMP.COD_COMP
LEFT JOIN factu_full_central..COMP_EMITIDOS AS COM
ON LIN.ASIEN_NUME = COM.ASIEN_NUME
AND COM.ESTADO != 'ANU'
LEFT JOIN factu_full_central..CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

```

WHERE

YEAR(CAB.FECHA) IN (2022,2023)

AND CAB.ASIEN_TIPO IN ('COB','FON','VEI','COI','VEN','MAN','MAI','STO')

AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
 '51216','51219','51214','51256','51257','51251','51226','51227',
 '51228','51229','21116','11112','11111','11132','11133','11131',
 '11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS

--AND (COM.ESTADO NOT IN ('CAN') OR COM.ESTADO IS NULL)

UNION ALL

-- RECIBOS DE SUELDO

SELECT 111111 AS
ASIEN_NUME,FECHA_ASIENTO,YYYYMM,LINEA_NUME,GRUPO,CUENTA,DESCRIP CUENTA,DESCRIP, TIPO CUENTA,
SUM(DEBE) AS DEBE, SUM(HABER) AS HABER, SUM(ACUMULADO) AS ACUMULADO,

```
COLUM,'' as OBSER, CONCAT(CUENTA,CODI_CENCO) AS CODI_CENCO, 'SUELDO' CONTROL_PRESUP, ASIEN_TIPO,
'' AS FACTURA_NRO, FECHA_FACTURA, 1111 AS CODIGO,'' as NOMBRE, COD_GRUPO, '' as AGRUPACION,'FRK' AS
EMPRESA
```

```
-----  
FROM ( SELECT  
    LIN.ASIEN_NUME,  
    CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,  
    CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))  
        ELSE CONCAT(YEAR(CAB.FECHA), '-', MONTH(CAB.FECHA))  
    END AS YYYYMM,  
    LIN.LINEA_NUME,  
    LEFT(LIN.CUENTA,4) AS GRUPO,  
    LIN.CUENTA,  
    PC.DESCRIP CUENTA,  
    LEFT(PC.DESCRIP CUENTA,4) AS DESCrip,  
    PC.TIPO CUENTA,  
    CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,  
    CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,  
    CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR  
        WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,  
    LIN.COLUM,  
    LIN.OBSER,  
    LIN.CODI_CENCO,  
    PC.CONTROL_PRESUP,  
    CAB.ASIEN_TIPO,  
    CONCAT (COMP.TIPO,'-',RIGHT (CONCAT ('0000000000',COMP.NUM),8)) AS FACTURA_NRO,  
    CONVERT(DATE,COMP.FECHA,103) AS FECHA_FACTURA,  
    COMP.PROVE AS CODIGO,  
    PRO.RAZON AS NOMBRE,  
    TCOMP.COD_GRUPO  
  
    FROM factu_full_central..ASIEN_LINEA AS LIN  
        LEFT JOIN factu_full_central..PLAN_CUENTAS AS PC  
        ON LIN.CUENTA = PC.CUENTA  
        LEFT JOIN factu_full_central..ASIEN_CABE AS CAB  
        ON LIN.ASIEN_NUME = CAB.ASIEN_NUME  
        LEFT JOIN factu_full_central..COMPRO_CABE_PROVE AS COMP  
        ON LIN.ASIEN_NUME = COMP.ASIEN_NUME  
        AND COMP.ESTADO <> 'ANU'  
        LEFT JOIN factu_full_central..PROVEEDORES AS PRO  
        ON COMP.PROVE = PRO.NUM_PROV  
        LEFT JOIN factu_full_central..COMP AS TCOMP  
        ON COMP.TIPO = TCOMP.COD_COMP  
        LEFT JOIN factu_full_central..COMP_EMITIDOS AS COM  
        ON LIN.ASIEN_NUME = COM.ASIEN_NUME  
        AND COM.ESTADO != 'ANU'  
        LEFT JOIN factu_full_central..CLIENTES AS CLI  
        ON COM.CLIENTE = CLI.NUM_CLIENTE
```

```

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('PRO','PAG','PRI','PAI','GEN','COB','PRI','PAI')
-----  

AND LIN.CUENTA IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311')
-----  

UNION ALL
SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
    ELSE CONCAT(YEAR(CAB.FECHA), '-' ,MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP_CUENTA,
LEFT(PC.DESCRIP_CUENTA,4) AS DESCrip,
PC.TIPO_CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
    WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
LIN.CODI_CENCO,
PC.CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FC_FACTURA_NRO,
CONVERT(DATE,COM.FECHA,103) AS FECHA_FACTURA,
COM.CLIENTE AS CODIGO,
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO

FROM factu_full_central..ASIEN_LINEA AS LIN
LEFT JOIN factu_full_central..PLAN CUENTAS AS PC
ON LIN.CUENTA = PC.CUENTA
LEFT JOIN factu_full_central..ASIEN_CABE AS CAB
ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
LEFT JOIN factu_full_central..COMPRO_CABE_PROVE AS COMP
ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
AND COMP.ESTADO <> 'ANU'
LEFT JOIN factu_full_central..PROVEEDORES AS PRO
ON COMP.PROVE = PRO.NUM_PROV
LEFT JOIN factu_full_central..COMP AS TCOMP
ON COMP.TIPO = TCOMP.COD_COMP
LEFT JOIN factu_full_central..COMP_EMITIDOS AS COM
ON LIN.ASIEN_NUME = COM.ASIEN_NUME

```

```

        AND COM_ESTADO != 'ANU'
        LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('COB','FON','VEI','COI','VEN','MAN','MAI','STO')

-----
AND LIN.CUENTA IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311')) AS T

-----
GROUP BY FECHA_ASIENTO,YYMM,LINEA_NUME,GRUPO,CUENTA,DESCRIP CUENTA,DESCrip,TIPO CUENTA,
COLUM,OBSER,CODI_CENCO, CONTROL_PRESUP, ASIEN_TIPO,FACTURA_NRO, FECHA_FACTURA,NOMBRE ,CODIGO,COD_GRUPO

UNION ALL

-- ACA PONGO DCB --

-----
-- FACTURAS DE VENTAS

-----
SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA),'-0',MONTH(CAB.FECHA))
      ELSE CONCAT(YEAR(CAB.FECHA),'-',MONTH(CAB.FECHA))
END AS YYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP CUENTA,
LEFT(PC.DESCRIP CUENTA,4) AS DESCrip,
PC.TIPO CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
      WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 1' CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COMP.TIPO,'-',RIGHT (CONCAT ('0000000000',COMP.NUM),8)) AS FACTURA_NRO,
CONVERT(DATE,COMP.FECHA,103) AS FECHA_FACTURA,
COMP.PROVE AS CODIGO,

```

```

PRO.RAZON AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 AS AGRUPACION,
'DCB' AS EMPRESA

FROM factu_full_dist_central..ASIEN_LINEA AS LIN
    LEFT JOIN factu_full_dist_central..PLAN CUENTAS AS PC
        ON LIN.CUENTA = PC.CUENTA
    LEFT JOIN factu_full_dist_central..ASIEN_CABE AS CAB
        ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
    LEFT JOIN factu_full_dist_central..COMPRO_CABE_PROVE AS COMP
        ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
        AND COMP.ESTADO <> 'ANU'
    LEFT JOIN factu_full_dist_central..PROVEEDORES AS PRO
        ON COMP.PROVE = PRO.NUM_PROV
    LEFT JOIN factu_full_dist_central..COMP AS TCOMP
        ON COMP.TIPO = TCOMP.COD_COMP
    LEFT JOIN factu_full_dist_central..COMP_EMITIDOS AS COM
        ON LIN.ASIEN_NUME = COM.ASIEN_NUME
        AND COM.ESTADO != 'ANU'
    LEFT JOIN factu_full_dist_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('PRO','PAG','PRI','PAI','GEN','COB','PRI','PAI')
-----  

AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
    '51216','51219','51214','51256','51257','51251','51226','51227',
    '51228','51229','21116','11112','11111','11132','11133','11131',
    '11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS

UNION ALL
-----  

-- ASIENTOS DE CMV
-----  

SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
    ELSE CONCAT(YEAR(CAB.FECHA), '-', MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP CUENTA,
LEFT(PC.DESCRIP CUENTA,4) AS DESCRIP,
PC.TIPO CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,

```

```

CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
      WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 2' CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COM.TIPO, '-' ,RIGHT (CONCAT ('0000000000', COM.NUM),8)) AS FACTURA_NRO,
--COM.NUM AS FACTURA_NRO,
CONVERT(DATE, COMP.FECHA, 103) AS FECHA_FACTURA,
COM.CLIENTE AS CODIGO, -- ACA PONGO EN CODIGO EL CAMPO OBSERVACIONE DEL VOUCHER PARA OBTENER EL NRO DE
CLIENTE
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 as AGRUPACION,
'DCB' AS EMPRESA

```

```

FROM factu_full_dist_central..ASIEN_LINEA AS LIN
LEFT JOIN factu_full_dist_central..PLAN CUENTAS AS PC
ON LIN.CUENTA = PC.CUENTA
LEFT JOIN factu_full_dist_central..ASIEN_CABE AS CAB
ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
LEFT JOIN factu_full_dist_central..COMPROMO_CABE_PROVE AS COMP
ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
AND COMP.ESTADO <> 'ANU'
LEFT JOIN factu_full_dist_central..PROVEEDORES AS PRO
ON COMP.PROVE = PRO.NUM_PROV
LEFT JOIN factu_full_dist_central..COMP AS TCOMP
ON COMP.TIPO = TCOMP.COD_COMP
LEFT JOIN factu_full_dist_central..COMP_EMITIDOS AS COM
ON ( LIN.ASIEN_NUME = COM.ASIEN_NUME and COM.ESTADO != 'ANU' )
LEFT JOIN factu_full_dist_central..CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

```

```

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('CMV')
-----
```

```

AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS

```

UNION ALL

-- PROVEEDORES

```

SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
      ELSE CONCAT(YEAR(CAB.FECHA), '-' ,MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP CUENTA,
LEFT(PC.DESCRIP CUENTA,4) AS DESCRIP,
PC.TIPO CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
      WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
CONCAT(LIN.CUENTA,LIN.CODI_CENCO) AS CODI_CENCO,
'PARTE 3' CONTROL_PRESUP,
CAB.ASIEN TIPO,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FC_FACTURA_NRO,
CONVERT(DATE,COM.FECHA,103) AS FECHA_FACTURA,
COM.CLIENTE AS CODIGO,
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO,
CLI.AGRU_3 as AGRUPACION,
'DCB' AS EMPRESA

FROM factu_full_dist_central..ASIEN_LINEA AS LIN
    LEFT JOIN factu_full_dist_central..PLAN CUENTAS AS PC
    ON LIN.CUENTA = PC.CUENTA
    LEFT JOIN factu_full_dist_central..ASIEN_CABE AS CAB
    ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
    LEFT JOIN factu_full_dist_central..COMPROMO_CABE_PROVE AS COMP
    ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
    AND COMP.ESTADO <> 'ANU'
    LEFT JOIN factu_full_dist_central..PROVEEDORES AS PRO
    ON COMP.PROVE = PRO.NUM_PROV
    LEFT JOIN factu_full_dist_central..COMP AS TCOMP
    ON COMP.TIPO = TCOMP.COD_COMP
    LEFT JOIN factu_full_dist_central..COMP_EMITIDOS AS COM
    ON LIN.ASIEN_NUME = COM.ASIEN_NUME
    AND COM.ESTADO != 'ANU'
    LEFT JOIN factu_full_dist_central..CLIENTES AS CLI
    ON COM.CLIENTE = CLI.NUM_CLIENTE

```

WHERE

YEAR(CAB.FECHA) IN (2022,2023)

```

AND CAB.ASIEN_TIPO IN ('COB','FON','VEI','COI','VEN','MAN','MAI','STO')
-----
AND LIN.CUENTA NOT IN ('51211','51217','51290','51213','51218','51215',
    '51216','51219','51214','51256','51257','51251','51226','51227',
    '51228','51229','21116','11112','11111','11132','11133','11131',
    '11134','21114','11115','11161','21111','51531','21311') -- CUENTAS SUELDO EXCLUIDAS
--AND (COM.ESTADO NOT IN ('CAN') OR COM.ESTADO IS NULL)
UNION ALL

-----
-- RECIBOS DE SUELDO

-----
SELECT 111111 AS
ASIEN_NUME,FECHA_ASIENTO,YYYYMM,LINEA_NUME,GRUPO,CUENTA,DESCRIP_CUENTA,DESCRIP,TIPO_CUENTA,
SUM(DEBE) AS DEBE,SUM(HABER) AS HABER,SUM(ACUMULADO) AS ACUMULADO,
COLUM,'' AS OBSER, CONCAT(CUENTA,CODI_CENCO) AS CODI_CENCO, 'SUELDO' CONTROL_PRESUP, ASIEN_TIPO,
'' AS FACTURA_NRO, FECHA_FACTURA, 1111 AS CODIGO,'' AS NOMBRE,COD_GRUPO, '' AS AGRUPACION,'DCB' AS
EMPRESA

-----
FROM (
    SELECT
        LIN.ASIEN_NUME,
        CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
        CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
            ELSE CONCAT(YEAR(CAB.FECHA), '-' ,MONTH(CAB.FECHA))
        END AS YYYYMM,
        LIN.LINEA_NUME,
        LEFT(LIN.CUENTA,4) AS GRUPO,
        LIN.CUENTA,
        PC.DESCRIP_CUENTA,
        LEFT(PC.DESCRIP_CUENTA,4) AS DESCRIPT,
        PC.TIPO_CUENTA,
        CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
        CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
        CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
            WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
        LIN.COLUM,
        LIN.OBSER,
        LIN.CODI_CENCO,
        PC.CONTROL_PRESUP,
        CAB.ASIEN_TIPO,
        CONCAT (COMP.TIPO,'-',RIGHT (CONCAT ('0000000000',COMP.NUM),8)) AS FACTURA_NRO,
        CONVERT(DATE,COMP.FECHA,103) AS FECHA_FACTURA,
        COMP.PROVE AS CODIGO,
        PRO.RAZON AS NOMBRE,
        TCOMP.COD_GRUPO

        FROM factu_full_dist_central..ASIEN_LINEA AS LIN
        LEFT JOIN factu_full_dist_central..PLAN_CUENTAS AS PC
        ON LIN.CUENTA = PC.CUENTA
        LEFT JOIN factu_full_dist_central..ASIEN_CAB AS CAB

```

```

    ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
    LEFT JOIN factu_full_dist_central..COMPRO_CABE_PROVE AS COMP
    ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
    AND COMP.ESTADO <> 'ANU'
    LEFT JOIN factu_full_dist_central..PROVEEDORES AS PRO
    ON COMP.PROVE = PRO.NUM_PROV
    LEFT JOIN factu_full_dist_central..COMP AS TCOMP
    ON COMP.TIPO = TCOMP.COD_COMP
    LEFT JOIN factu_full_dist_central..COMP_EMITIDOS AS COM
    ON LIN.ASIEN_NUME = COM.ASIEN_NUME
    AND COM.ESTADO != 'ANU'
    LEFT JOIN factu_full_dist_central..CLIENTES AS CLI
    ON COM.CLIENTE = CLI.NUM_CLIENTE

```

```

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('PRO','PAG','PRI','PAI','GEN','COB','PRI','PAI')

```

```

AND LIN.CUENTA IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311')

```

```

UNION ALL
SELECT
LIN.ASIEN_NUME,
CONVERT(DATE,CAB.FECHA,103) AS FECHA_ASIENTO,
CASE WHEN LEN(MONTH(CAB.FECHA)) = 1 THEN CONCAT(YEAR(CAB.FECHA), '-0', MONTH(CAB.FECHA))
ELSE CONCAT(YEAR(CAB.FECHA), '-', MONTH(CAB.FECHA))
END AS YYYYMM,
LIN.LINEA_NUME,
LEFT(LIN.CUENTA,4) AS GRUPO,
LIN.CUENTA,
PC.DESCRIP CUENTA,
LEFT(PC.DESCRIP CUENTA,4) AS DESCrip,
PC.TIPO CUENTA,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR END AS DEBE,
CASE WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS HABER,
CASE WHEN LIN.COLUM = 'H' THEN LIN.IMPOR
WHEN LIN.COLUM = 'D' THEN LIN.IMPOR * -1 END AS ACUMULADO,
LIN.COLUM,
LIN.OBSER,
LIN.CODI_CENCO,
PC.CONTROL_PRESUP,
CAB.ASIEN_TIPO,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FC_FACTURA_NRO,
CONVERT(DATE,COM.FECHA,103) AS FECHA_FACTURA,
COM.CLIENTE AS CODIGO,
CLI.NOM_FANTASIA AS NOMBRE,
TCOMP.COD_GRUPO

```

```

FROM factu_full_dist_central..ASIEN_LINEA AS LIN
LEFT JOIN factu_full_dist_central..PLAN_CUENTAS AS PC
ON LIN.CUENTA = PC.CUENTA
LEFT JOIN factu_full_dist_central..ASIEN_CABE AS CAB
ON LIN.ASIEN_NUME = CAB.ASIEN_NUME
LEFT JOIN factu_full_dist_central..COMPRO_CABE_PROVE AS COMP
ON LIN.ASIEN_NUME = COMP.ASIEN_NUME
AND COMP.ESTADO <> 'ANU'
LEFT JOIN factu_full_dist_central..PROVEEDORES AS PRO
ON COMP.PROVE = PRO.NUM_PROV
LEFT JOIN factu_full_dist_central..COMP AS TCOMP
ON COMP.TIPO = TCOMP.COD_COMP
LEFT JOIN factu_full_dist_central..COMP_EMITIDOS AS COM
ON LIN.ASIEN_NUME = COM.ASIEN_NUME
AND COM.ESTADO != 'ANU'
LEFT JOIN factu_full_dist_central..CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

```

```

WHERE YEAR(CAB.FECHA) IN (2022,2023)
AND CAB.ASIEN_TIPO IN ('COB','FON','VEI','COI','VEN','MAN','MAI','STO')

```

```

AND LIN.CUENTA IN ('51211','51217','51290','51213','51218','51215',
'51216','51219','51214','51256','51257','51251','51226','51227',
'51228','51229','21116','11112','11111','11132','11133','11131',
'11134','21114','11115','11161','21111','51531','21311')) AS T

```

```

GROUP BY FECHA_ASENTO, YYYYMM, LINEA_NUME, GRUPO, CUENTA, DESCRIP CUENTA, DESCrip, TIPO CUENTA,
COLUM, OBSER, CONCAT(CUENTA, CODI_CENCO), CONTROL_PRESUP, ASIEN_TIPO, FACTURA_NRO, FECHA_FACTURA, NOMBRE
,CODIGO, COD_GRUPO

```

Vista - PNL_Ventas_Sucursales

```
SELECT
ITM.TIPO_FACT,
TIPO_COMP.DESCRIP,
TIPO_COMP.COD_GRUPO as DEPO_COD,
ITM.DEPO as DEPOSITO,
CONCAT (ITM.TIPO_FACT, ' - ', RIGHT (CONCAT (' 0000000000', ITM.NUM_FACT), 8)) AS FACTURA_NRO,
convert(date, COM.FECHA, 13) AS FECHA,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA), ' - ', MONTH(COM.FECHA))
END AS YYYYMM,
COM.CUIT_COMPRO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CLI.AGRU_3,
ITM.ARTICULO,
ART.DESCRIP_ARTI,
ITM.CANT * -1 AS CANT,
ITM.PRECIO,
(ITM.CANT * Itm.PRECIO * -1) - (ITM.CANT * PRECIO_NETO * (1+ ITM.PORCEN_IVA/100) * -1) AS
IMPORTE_DESCUENTO,
CASE WHEN TIPO_COMP.PREFIJO <> '0000' -- EN LOCALES SE USA EL PREFIJO PARA SABER SI ES FISCAL O NO
     THEN ITM.CANT * PRECIO_NETO * -1.21
     ELSE ITM.CANT * PRECIO_NETO * -1 END AS FACTURADO_TOTAL_C_IVA,
CASE WHEN TIPO_COMP.PREFIJO <> '0000'
     THEN ITM.CANT * PRECIO_NETO * -1
     ELSE ITM.CANT * PRECIO_NETO / -1.21 END AS FACTURADO_TOTAL_S_IVA,
CASE WHEN TIPO_COMP.PREFIJO <> '0000'
     THEN ITM.CANT * ITM.COSTO_UNI * 1.21
     ELSE ITM.CANT * ITM.COSTO_UNI * 1.21 END AS CMV_C_IVA,
CASE WHEN TIPO_COMP.PREFIJO <> '0000'
     THEN ITM.CANT * ITM.COSTO_UNI
     ELSE ITM.CANT * ITM.COSTO_UNI END AS CMV_S_IVA,
CASE WHEN TIPO_COMP.PREFIJO <> '0000'
     THEN ITM.CANT * ITM.CST_CTRL * 1.21
     ELSE ITM.CANT * ITM.CST_CTRL * 1.21 END AS CMV_CTRL_C_IVA,
CASE WHEN TIPO_COMP.COD_GRUPO = '0000'
     THEN ITM.CANT * ITM.CST_CTRL
     ELSE ITM.CANT * ITM.CST_CTRL END AS CMV_CTRL_S_IVA,
COM.ESTADO,
AGR.DESCRIP_AGRU AS GRUPO,
CASE WHEN SUBSTRING ( ITM.ARTICULO, 1, 2) = 'PR' THEN SUBSTRING ( ITM.ARTICULO, 1, 2)
     ELSE SUBSTRING ( ITM.ARTICULO, 1, 1) END as TIPO_PRODUCTO,
ITM.COLOR AS PARTIDA,
```

```

CONVERT(date, DATEADD(DD, -1, DATEADD(MM, 0, GETDATE()))), 3) AS FECHA_Reporte,
CASE WHEN COD_GRUPO = 'FRS' THEN '41300FL'
      WHEN COD_GRUPO = 'MAL' THEN '41300VC'
      WHEN COD_GRUPO = 'BEL' THEN '41300BE'
      WHEN COD_GRUPO = 'PAL' THEN '41300PA'
      WHEN COD_GRUPO = 'CER' THEN '41300CE'
      WHEN COD_GRUPO = 'MRS' THEN '41300MS'
      WHEN COD_GRUPO = 'LHE' THEN '41300RE'
      WHEN COD_GRUPO = 'FR1' THEN '41113'
      WHEN COD_GRUPO = 'FBE' THEN '41113'
      WHEN COD_GRUPO = 'FPA' THEN '41113'
      WHEN COD_GRUPO = 'FFL' THEN '41113'
      WHEN COD_GRUPO = 'MST' THEN '41300MS'
      ELSE 'XX' END CUENTA_Revenue,
CASE WHEN COD_GRUPO = 'FRS' THEN '51300FL'
      WHEN COD_GRUPO = 'MAL' THEN '51300VC'
      WHEN COD_GRUPO = 'BEL' THEN '51300BE'
      WHEN COD_GRUPO = 'PAL' THEN '51300PA'
      WHEN COD_GRUPO = 'CER' THEN '51300CE'
      WHEN COD_GRUPO = 'MRS' THEN '51300MS'
      WHEN COD_GRUPO = 'LHE' THEN '51300RE'
      WHEN COD_GRUPO = 'FR1' THEN '51113'
      WHEN COD_GRUPO = 'FBE' THEN '51113'
      WHEN COD_GRUPO = 'FPA' THEN '51113'
      WHEN COD_GRUPO = 'FFL' THEN '51113'
      WHEN COD_GRUPO = 'MST' THEN '51300MS'
      ELSE 'XX' END CUENTA_Cost,
COM.CODI_VENDE,
CASE WHEN CLI.AGRU_3 = 'RAP' THEN (ITM.CANT * -1 * ITM.PRECIO_NETO *1.21 * 0.15 ) * 1.21
      WHEN CLI.AGRU_3 = 'PEDY' THEN (ITM.CANT * -1 * ITM.PRECIO_NETO *1.21 * 0.14 ) * 1.21
END AS COMISION_APP

```

```

FROM factu_full_locales..RENG_FAC AS ITM LEFT JOIN factu_full_locales..COMP_EMITIDOS AS COM
ON ITM.NUM_FACT = COM.NUM AND ITM.TIPO_FACT = COM.TIPO
LEFT JOIN factu_full_locales..ARTICULOS AS ART
ON ITM.ARTICULO = ART.COD_ARTICULO
LEFT JOIN factu_full_locales..CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE
LEFT JOIN factu_full_locales..AGRUPACIONES AS AGR
ON ART.AGRU_1= AGR.CODI_AGRU
AND AGR.NUM_AGRU= 1
LEFT JOIN factu_full_locales..COMP AS TIPO_COMP
ON ITM.TIPO_FACT = TIPO_COMP.COD_COMP
-- LEFT JOIN full_powerbi_kpi_empresa_depositos AS DEP
-- ON ITM.DEP0 = DEP.DEP0_CODI

```

```
WHERE YEAR(COM.FECHA) IN ('2022','2023')
AND MONTH(COM.FECHA) in ( '01','02','03','04','05','06','07','08','09','10','11','12')
AND (ITM.TIPO_FACT LIKE ('F%') OR
    ITM.TIPO_FACT LIKE ('C%') OR
    ITM.TIPO_FACT LIKE ('B%')) )
--AND ITM.TIPO_FACT NOT IN ('FCF') -- elimino la fc de conformidad
AND ITM.DEP0 IN ('MAL','FRS','BEL','PAL','CER','MRS','MST','LHE')
AND ITM.COMPO_ITEM IS NULL -- SE SUMO PARA QUE NO TRAIGA EL DETALLE DE LOS COMPONENTES
AND COM.USER_ALTA not like 'fel_%_mayorista' -- 16-04-2024 Ailen. Se agregó para que no se muestre en el
reporte la venta mayorista
```

Consideraciones

- Cuando se crea una nueva sucursal propia, es necesario poner en el WHEN COD_GRUPO la condición del Deposito y la cuenta contable de facturacion y de cmv
- Ademas es necesario sumar el deposito en el Where

T-pbi_stock_depositos

Es una tabla que se genera todos los últimos días del mes para guardar la posición de stock. Se corre desde un proceso en central automáticamente

SQL de la extracción

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

T-pbi_sucursales_canales

/Se usa para asignar un tipo de cliente en la base de locales al canal correspondiente.

```
[NUM_CLIENTE] [char](5) NULL,  
[NOM_FANTASIA] [varchar](20) NULL,  
[AGRU_3] [varchar](6) NULL,  
[CANAL] [varchar](10) NULL
```

Tabla

	NUM_CLIENTE	NOM_FANTASIA	AGRU_3	CANAL	
1	1		CANJE	ON LINE	
2	2	Globo	GLO	ON LINE	
3	3	Rappi	RAP	ON LINE	
4	4	PEDIDOS YA	PEYA	ON LINE	
5	5	UBER	UBER	ON LINE	
6	6	WABI	WABI	ON LINE	
7	7	Mercado Libre		ON LINE	
8	8	Mercado Envio	MELI	ON LINE	
9	50	Ventas Web Ecommerce	VWE	ON LINE	

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

T-pbi_ventas_locales

Esta tabla se actualiza todas las noches desde la base de reportes por medio del siguiente SP
pbi_ventas_locales_insert

```
USE [factu_full_reportes]
GO
/***** Object:  StoredProcedure [dbo].[pbi_ventas_locales_insert]      Script Date: 24/06/2024 09:53:14
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[pbi_ventas_locales_insert]
as

truncate table pbi_ventas_locales

insert into pbi_ventas_locales (
    FACTURA_NRO ,FECHA_COMPLETA ,
    HORA , FECHA , CLIENTE , CANAL ,
    DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
    AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
    VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
    TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
    COMPO_ITEM , RENGLON , FECHA_Reporte , BASE_DATOS,CUENTA_CONTABLE )
select FACTURA_NRO ,FECHA_COMPLETA ,
    HORA , FECHA , CLIENTE , CANAL ,
    DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
    AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
    VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
    TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
    COMPO_ITEM , RENGLON , FECHA_Reporte , 'factu_full_locales',
    CUENTA_CONTABLE
from factu_full_locales..[PBI.Ventas]

insert into pbi_ventas_locales(
    FACTURA_NRO ,FECHA_COMPLETA ,
    HORA , FECHA , CLIENTE , CANAL ,
    DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
    AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
    VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
    TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
    COMPO_ITEM , RENGLON , FECHA_Reporte , BASE_DATOS, CUENTA_CONTABLE )
```

```
select FACTURA_NRO ,FECHA_COMPLETA ,
       HORA , FECHA , CLIENTE , CANAL ,
       DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
       AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
       VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
       TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
       COMPO_ITEM , RENGLON , FECHA_Reporte , 'factu_full_bahia_blanca',
       CUENTA_CONTABLE
from factu_full_bahia_blanca..pbi_ventas
```

```
insert into pbi_ventas_locales(
FACTURA_NRO ,FECHA_COMPLETA ,
       HORA , FECHA , CLIENTE , CANAL ,
       DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
       AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
       VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
       TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
       COMPO_ITEM , RENGLON , FECHA_Reporte, BASE_DATOS, CUENTA_CONTABLE )
select FACTURA_NRO ,FECHA_COMPLETA ,
       HORA , FECHA , CLIENTE , CANAL ,
       DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
       AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
       VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
       TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
       COMPO_ITEM , RENGLON , FECHA_Reporte , 'factu_full_central',
       CUENTA_CONTABLE
from factu_full_central..pbi_ventas
```

```
insert into pbi_ventas_locales(
FACTURA_NRO ,FECHA_COMPLETA ,
       HORA , FECHA , CLIENTE , CANAL ,
       DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
       AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
       VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
       TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
       COMPO_ITEM , RENGLON , FECHA_Reporte, BASE_DATOS, CUENTA_CONTABLE )
select FACTURA_NRO ,FECHA_COMPLETA ,
       HORA , FECHA , CLIENTE , CANAL ,
       DEPO , USER_ALTA , ARTICULO , DESCRIP_ARTI ,
       AGRU_1 , DESCRIP_AGRU , PROMO , CANT , PRECIO ,
       VENTA_C_IVA , CMV_C_IVA , VENTA_S_IVA , CMV_S_IVA ,
       TIPO_PRODUCTO , TICKET_QTY , CANT_COMBO ,
       COMPO_ITEM , RENGLON , FECHA_Reporte , 'factu_full_dist_central',
       CUENTA_CONTABLE
from factu_full_dist_central..pbi_ventas
```

El SP usa 3 vistas en cada uno de los ambientes

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

Para los locales, es necesario cambiar las tablas que estan generandose en las sucursales free

```
adm_buscar_string 'full_PBI_Ventas_union'  
Sp_HELPTEXT full_PBI_Ventas_union
```

T-pbi_sucursales_tipo_producto

Diccionario

```
[Tipo_Producto] [varchar](5) NULL,  
[Descripcion] [varchar](50) NULL
```

	Tipo_Producto	Descripcion
1	C	Producto Central
2	L	Producto Local
3	LC	Producto Local
4	PR	Promocion
5	P	Publicitario
6	B	Combo

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	

Consideraciones

T-pbi_calendario

Es una tabla que se genera todos los primeros días del mes desde un scrip dentro del servidor de Fullescabio ejecutando el SP pbi_calendario_update

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-8984)	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11754)	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-13654)	

Consideraciones

V-full_powerbi_KPI_empresa_cantidad_sku

Desde la vista de ventas, busca y cuenta todos los productos de una categoria para hacer el promedio, usando solo los canales habilitados y tipo de productos que comiencen con la letra C, P, B y M dentro de los meses que estan en la tabla calendario Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11634>)

```
select t.AGRU_1 , COUNT (t.articulo) as Cant_SKU  
  
from (  
    SELECT distinct v.ARTICULO, v.AGRU_1  
        FROM pbi_ventas_locales as v left JOIN pbi_calendario as c  
    on v.FECHA = c.YYYYMM  
    left join full_powerbi_kpi_empresa_clientes as cl  
    on v.CLIENTE = cl.NUM_Cliente  
    where c.POSICION between 1 and 6  
    and SUBSTRING (v.articulo,1,1) in ('C','P','B','M')  
    and cl.CODI_AGRU in ('full', 'prop','MELI','VWE','VINO','PERS')) as T  
  
group by t.AGRU_1
```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	

T-pbi_brechas_rotacion

Usamos esta tabla para definir cual es el Gap que se va a usar desde el valor medio de una categoría para determinar el valor de Alta y Baja

Por ejemplo, si la categoría tiene un valor medio de consumo en unidades de 300 y el porcentaje_Brechas es del 30% la cuenta que hacemos es la siguiente

$$\text{Alta} = 300 * (1+0.3) \square 390$$

$$\text{Baja} = 300 * (1+0.3) \square 230$$

lo que significa que los valores por arriba de 390 son de alta rotacion, los de menos de 230 de baja y los comprendidos entre 390 y 230 son media.

Para cambiar la brecha de una categoría se debe ejecutar un update de la tabla

Tabla

```
select *
from pbi_brechas_rotacion
```

	AGRU_1	DESCRIP_AGRU	PROCENTAJE_BRECHAS
1	1001	Accesorios	0.20
2	1002	Aguardientes	0.20
3	1003	Aperitivos	0.20
4	1004	Cervezas Lata	0.20
5	1005	Cervezas Litro	0.20
6	1006	Cervezas Porron	0.20
7	1007	Champagnes	0.20
8	1008	Combos	0.20
9	1009	Delicatessen	0.20
10	1012	Espumantes	0.20
11	1013	Gin	0.20
12	1014	Heladera	0.20
13	1017	Kits Mercado Libre	0.20

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	

Consideraciones

SQL para sumar una categoria

```
insert into pbi_brechas_rotacion (AGRU_1,DESCRIP_AGRU,PROCENTAJE_BRECHAS)
values('1037','Gastos Logisticos',0.20);
```

SQL para updatear una categoria

V- full_powerbi_KPI_empresa_cantidad_vendida_skus

Desde la vista de ventas, busca y suma todas las cantidades de los productos de una categoria para hacer el promedio, usando solo los canales habilitados y tipo de productos que comiencen con la letra C, P, B y M dentro de los meses que estan en la tabla calendario Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11634>)

```
SELECT AGRU_1, sum(cant) as Cant  
  
FROM pbi_ventas_locales as v left JOIN pbi_calendario as c  
on v.FECHA = c.YYYYMM  
left join full_powerbi_kpi_empresa_clientes as cl  
on v.CLIENTE = cl.NUM_Cliente  
  
where v.BASE_DATOS  = 'factu_full_central'  
and c.POSICION between 1 and 6  
and cl.CODI_AGRU in ('full', 'prop','MELI','VWE','VINO','PERS')  
  
group by AGRU_1
```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	

T-pbi_rotacion_historica

Es una tabla que se genera todos los últimos días del mes para guardar la rotación de los productos. Se corre desde un proceso en central automáticamente

Usa la extraccion que surge del concepto de rotacion para llegarse

SQL de la extracción

```
EXEC = [dbo].[pbi_rotacion_historica_insert]
```

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554)	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11934)	

Consideraciones

@Augusto por favor termina la documentación con el proceso automático

V-

full_powerbi_KPI_Logistica_estados_considerar

Se usa para determinar cuales son los estados de los pedidos que se deben considerar.

```
select DISTINCT ITM.CODI_ESTA,
       case when ITM.CODI_ESTA IN ( 'SS','REC','GES') THEN 'N' ELSE 'S' END AS 'CONSIDERAR'
  from factu_full_central.dbo.PEDI_ITEMS  as ITM
```

Toma el estado de la tabla depedidos Items y pone que NO a los que estan dentro del IN en el case when para todos los demas, se pone SI

Reporte que usa la vista	Link

V-full_powerbi_KPI_Logistica_Comprobantes

Se usa para determinar cuales son los comprobantes de los pedidos que se deben considerar.
En grupo ponemos los tipos de comprobantes y con eso determinamos el Grupo al que pertenece.
Todos los que estan arriba, los ponemos para considerar dentro de la extraccion

```
SELECT COD_COMP, DESCRIP,
CASE WHEN COD_COMP IN ('DML','DEC','DMA','DAC','DBL','DFU') THEN 'eCommerce'
     WHEN COD_COMP IN ('DDM','DDN','DDR','DDI') THEN 'Mayorista'
     WHEN COD_COMP IN ('DDW','DLP','DDX','DWN','DDD','DBI','DPK') THEN 'Venta'
     ELSE 'VER' END AS GRUPO,
CASE WHEN COD_COMP IN ('DAC','DBL','DEC','DMA','DML','DFU','DDI',
                      'DDM','DDN','DDR','DBI','DDD','DDW','DDX','DLP',
                      'DPK','DWN','DLL') THEN 'S' ELSE 'N' END AS 'CONSIDERAR'
FROM factu_full_central.dbo.COMP
WHERE SUBSTRING (COD_COMP,1,1) = 'D'
```

Reporte que usa la vista	Link

V-

full_powerbi_KPI_Logistica_Articulos_considerar

Se usa para determinar cuales son los articulos que se deben considerar dentro del conteo de productos en la estadística, excluimos los costo de envíos por ejemplo

```
select COD_ARTICULO,  
       case when substring (cod_articulo,1,1) IN ( 'A') THEN 'N' ELSE 'S' END AS 'CONSIDERAR'  
  from factu_full_central.dbo.ARTICULOS as art
```

Reporte que usa la vista	Link

V-full_powerbi_KPI_Sucursales_Clientes

```
select NUM_CLIENTE, RAZON, DOMICILIO , TELEFONO , LOCALIDAD , CUIT, MAIL, FECHA_ALTA , PROVINCIA,  
USER_ALTA  
from factu_full_locales.dbo.clientes  
where num_cliente > 100  
and IVA > 1
```

Listado de todas las marcas agregadas en la base de locales

Reporte que usa la vista	Link
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-16394)	

Documentacion Store Procedure

P - pbi_rotacion_historica_insert

Usando el query establecido por el Concepto de rotación este Store Procedure inserta en la tabla de pbi_rotacion_historica los datos. La columna YYYYMM se completa con la fecha en que se ejecuto el SP.

```
insert into pbi_rotacion_historica(Rotacion ,Cod_Articulo ,Categoria ,YYYYMM ,Cat_Total_ventas_Semestral  
,Cat_Total_ventas_mensual ,Cat_Cant_SKU ,SKU_Total_mensual ,Alta ,Baja)  
SELECT  
    case  
        when round(sum(v.cant) / 6,2) > cv.cant/ 6 / w.cant_sku * (1+r.PROCENTAJE_BRECHAS)      then 'Alta'  
        when round(sum(v.cant) / 6,2) < cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS) then 'Baja'  
        else 'Media' end as Rotacion,  
    v.ARTICULO , v.AGRU_1,  
    replace(left(convert(varchar,getdate(),23),7),'-','') as YYYYMM,  
    round(sum(v.cant),2) as Cat_Total_ventas_Semestral,  
    round(sum(v.cant) / 6,2) as Cat_Total_ventas_mensual,  
    w.cant_sku as Cat_Cant_SKU,  
    cv.cant/ 6 / w.cant_sku as SKU_Total_mensual,  
    cv.cant/ 6 / w.cant_sku * (1+r.PROCENTAJE_BRECHAS)   as Alta,  
    cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS)   as Baja  
  
FROM pbi_ventas_locales as v left JOIN pbi_calendario as c  
on v.FECHA = c.YYYYMM  
left join full_powerbi_kpi_empresa_clientes as cl  
on v.CLIENTE = cl.NUM_Cliente  
left join full_powerbi_KPI_empresa_cantidad_sku as w  
on v.AGRU_1 = w.agru_1  
left join pbi_brechas_rotacion r  
on v.AGRU_1 = r.AGRU_1  
left join full_powerbi_KPI_empresa_cantidad_vendida_sku as cv  
on v.AGRU_1 = cv.AGRU_1  
  
where v.BASE_DATOS  = 'factu_full_central'  
and c.POSICION between 1 and 6  
and cl.CODI_AGRU in ('full', 'prop','MELI','VWE','VINO','PERS')  
  
group by v.ARTICULO , v.AGRU_1, cv.cant/ 6 / w.cant_sku / (1+r.PROCENTAJE_BRECHAS), cv.cant/ 6 /  
w.cant_sku * (1+r.PROCENTAJE_BRECHAS),  
w.cant_sku, cv.cant/ 6 / w.cant_sku, r.PROCENTAJE_BRECHAS
```

Reporte que usa la vista

Link

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11554>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11834>)

P - pbi_calendario_update

Creado a pedido de Hernan, se toma YYYYMM y se agrega una numeracion siendo 0 el mes actual

.

SQL

```
DECLARE @MinDate DATE = '20220104',
        @MaxDate DATE = getdate(),
        @primera_fecha varchar(7),
        @posi_num int;

delete dbo.pbi_calendario;

with se as (
SELECT TOP (DATEDIFF(DAY, @MinDate, @MaxDate) + 1)
    Date = DATEADD(DAY, ROW_NUMBER() OVER(ORDER BY a.object_id) - 1, @MinDate)
FROM sys.all_objects a
CROSS JOIN sys.all_objects b
)
insert into dbo.pbi_calendario (yyyymm) select distinct left(date,7) YYYYMM from se

select top 1 @primera_fecha = yyyymm from dbo.pbi_calendario order by yyyymm desc

update dbo.pbi_calendario set posicion = 0 where yyyymm = @primera_fecha

DECLARE @RowCount INT = (SELECT COUNT(*) FROM dbo.pbi_calendario), @fecha varchar(7);

WHILE @RowCount > 0 BEGIN

SELECT @fecha = yyyymm FROM dbo.pbi_calendario
ORDER BY yyyymm asc OFFSET @RowCount - 1 ROWS FETCH NEXT 1 ROWS ONLY;

select @posi_num = max(posicion)+1 from dbo.pbi_calendario
update dbo.pbi_calendario set posicion = @posi_num where yyyymm = @fecha and yyyymm <> @primera_fecha

SET @RowCount -= 1;
END
```

Se declaran las variables y se vacia la tabla

```
DECLARE @MinDate DATE = '20220104',
        @MaxDate DATE = getdate(),
        @primera_fecha varchar(7),
```

```

@posi_num int;

delete dbo.pbi_calendario;

with se as (
SELECT TOP (DATEDIFF(DAY, @MinDate, @MaxDate) + 1)
    Date = DATEADD(DAY, ROW_NUMBER() OVER(ORDER BY a.object_id) - 1, @MinDate)
FROM sys.all_objects a
CROSS JOIN sys.all_objects b
)
insert into dbo.pbi_calendario (yyyymm) select distinct left(date,7) YYYYMM from se

```

Elegimos la fecha actual como primera fecha y agregamos el valor 0

```

select top 1 @primera_fecha = yyyymm from dbo.pbi_calendario order by yyyymm desc
update dbo.pbi_calendario set posicion = 0 where yyyymm = @primera_fecha

```

Creamos un proceso que agrega el resto de la numeracion a partir de la fecha

```

DECLARE @RowCount INT = (SELECT COUNT(*) FROM dbo.pbi_calendario), @fecha varchar(7);

WHILE @RowCount > 0 BEGIN

SELECT @fecha = yyyymm FROM dbo.pbi_calendario
ORDER BY yyyymm asc OFFSET @RowCount - 1 ROWS FETCH NEXT 1 ROWS ONLY;

select @posi_num = max(posicion)+1 from dbo.pbi_calendario
update dbo.pbi_calendario set posicion = @posi_num where yyyymm = @fecha and yyyymm <> @primera_fecha

SET @RowCount -= 1;
END

```

Reporte que usa la vista	
Private (https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-11634)	

pbi_DisplayOrder

Tabla creada por orden de Hernan

Está en el PBI en un campo creado.

Si querés incorporarlo a la tabla, es un SELECT según el valor de "CuentaDescripcion"

- = *Sucursales* ==> 1
- = *Mayorista* ==> 2
- = *contiene "Mayorista - Vende"* ==>3
- = *Mercado Libre* ==> 7
- = *Mercado Libre Full* ==> 6
- = *Web Ecommerce* ==> 8
- = *Vinotecas* ==>4
- = *Personal Fullescabio* ==>5
- = *Belgrano* ==> 9
- = *Cerviñ0* ==> 10
- = *Floresta* ==> 11
- = *Malabia* ==> 12
- = *Palermo* ==> 13
- = *ELSE* 0

Llamalo DisplayOrder

PBI - PNL

vistas

PNL

full_powerbi_kpi_empresa_presupuesto □ vista

full_powerbi_KPI_Sucursales_Comprobantes

Tabla

pbi_cuentas

pbi_cuentas_grupo

pbi_cuentas_subgrupo

pbi_jerarquias

pbi_ventas_feriados

PBI - Reporte General

Vistas

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6824>)

full_powerbi_kpi_empresa_articulos
full_powerbi_KPI_Sucursales_Sucursales
full_powerbi_KPI_Sucursales_Subcategorias
full_powerbi_KPI_Sucursales_Productos
full_powerbi_KPI_Sucursales_Componentes
full_powerbi_KPI_Sucursales_Marcas
full_powerbi_KPI_Sucursales_Categorias
full_powerbi_KPI_Sucursales_Comprobantes

Tabla

pbisucursales_canales
pbisucursales_tipo_producto
pbiventas_locales

PBI - Cobranzas

Vistas

full_powerbi_kpi_empresa_cobros

full_powerbi_kpi_empresa_cobros_mayoristas

full_powerbi_kpi_empresa_clientes

full_powerbi_kpi_empresa_ingresos_prom_pos

Tabla

PBI - Sellout

Vistas

PNL

full_powerbi_kpi_empresa_presupuesto

full_powerbi_kpi_empresa_cuentas_resultado

full_powerbi_kpi_empresa_clientes

full_powerbi_KPI_Sucursales_Comprobantes

Tabla

pbi_ventas_2021

pbi_ventas_feriados

pbi_DisplayOrder

Updates de nuevas sucursales propias

update pbi_DisplayOrder set DisplayOrder = ('16') where CuentaDescripcion = 'Sucursal Viii'

update pbi_DisplayOrder set CuentaDescripcion = ('Sucursal Congreso') where DisplayOrder = ('16')

full_powerbi_kpi_empresa_cuentas_resultado □ poner la cuenta de venta del comprobante en el
where

cambios

- Comparacion CM vs PY al dia de la fecha
- Abrir las sucursales propias en otra solapa
-

PBI - Reporte Sucursal

Vistas

full_powerbi_kpi_empresa_prov_articulo
full_powerbi_kpi_empresa_articulos
full_powerbi_KPI_Sucursales_Sucursales
full_powerbi_KPI_Sucursales_Subcategorias
full_powerbi_KPI_Sucursales_Productos
full_powerbi_KPI_Sucursales_Componentes
full_powerbi_KPI_Sucursales_Marcas
full_powerbi_KPI_Sucursales_Categorias
full_powerbi_KPI_Sucursales_Comprobantes

Tabla

pbisucursales_canales
pbisucursales_tipo_producto
pbiventas_locales

PBI - Sellin - Sellout

David va a usar este PBI para revisar las ventas de y hacia las sucursales buscando oportunidades de venta por categoría

Vistas

PNL □ no se usa hasta que la venta este aperturada por producto, mientras tanto vamos usar **full_powerbi_kpi_empresa_ventas_det**

Modificación

04/08/2023 □ se quito de la vista el campo AND ITM.COMPO_ITEM IS NULL para poder comparar unidades en si, y sumar las que están en combo.

full_powerbi_KPI_Sucursales_Categorias □ categorias de articulos

full_powerbi_kpi_empresa_articulos □ Articulos, toma el dato de la categoria y lo suma a las ventas de sellin

Tabla

pbi_ventas_locales □ Se usa para ver las ventas del sellout

pbi_relaciones □ se usa para relacionar los clientes de central con todas las razones sociales con todos los PDV de locales unificados por relacion

Consideraciones

- No se tienen que tomar los L para las comparaciones
- Los Compuestos deben ser contados por componentes
- Hay clientes que en el Sellin tienen una sola razon social y compran por depositos, armar una tabla que informe esta condición
- La venta de promos debe estar por unidades, ver como afecta a los pesos
-

prueba

categorias □

Vista - full_powerbi_kpi_empresa_cobros

----- GRUPO A - FACTURAS COMPLETAMENTE PAGADAS

```
SELECT 'FULL' AS EMPRESA,
CONCAT (CA.APLI_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.APLI_NUM ),8)) AS FACTURA_NRO,
COM.ESTADO,
COM.ESTADO AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA), '-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0', MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0', MONTH(CE.FECHA))
     ELSE CONCAT(YEAR(CE.FECHA), '-',MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA,CE.FECHA) AS DIAS,
0 AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA,CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
      END AS GAP_COBRANZAS,
'PAGADO' AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA,CE.FECHA) <=DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
'NO' END AS AUDITORIA,
COM.OBSER
```

```

FROM factu_full_central..COMPRO_API CA
    INNER JOIN factu_full_central..COMP C
        ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
        ON CA.TIPO = CE.TIPO
        AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
        ON COM.TIPO = CA.API_TIPO
        AND COM.NUM = CA.API_NUM
    LEFT JOIN factu_full_central..CLIENTES AS CLI
        ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 = 'FULL'
AND CONCAT (CA.API_TIPO,'-',RIGHT (CONCAT ('0000000000',CA.API_NUM ),8)) IN
    (SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
    FROM factu_full_central..COMP_EMITIDOS AS COM
    WHERE COM.TIPO LIKE 'F%'
    AND COM.ESTADO = 'CAN'
    AND COM.TOTAL <> 0
    AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())))
)

```

UNION ALL

----- GRUPO B - FACTURAS SIN PAGOS EFECTUADOS

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO,'-TOT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,

```

```

CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'
AND COM.SALDO = COM.TOTAL

```

```

UNION ALL
----- GRUPO C.0 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO LOS PAGOS
-----
```

```

SELECT 'FULL' AS EMPRESA,
CONCAT (CA.API_TIP0, '-' ,RIGHT (CONCAT ('0000000000', CA.API_NUM ),8)) AS FACTURA_NRO,
'CAN' AS ESTADO,
CONCAT(COM.ESTADO, '-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
CE.FECHA AS FECHA_COBRANZA,
DATEPART(DW, COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK, COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK, COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK, CE.FECHA) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA), '-0', MONTH(COM.FECHA))

```

```

        ELSE CONCAT(YEAR(COM.FECHA), '-' ,MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO), '-0' ,MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO), '-' ,MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(CE.FECHA)) = 1 THEN CONCAT(YEAR(CE.FECHA), '-0' ,MONTH(CE.FECHA))
     ELSE CONCAT(YEAR(CE.FECHA), '-' ,MONTH(CE.FECHA)) END AS YYYYMM_PAGO_REAL,
CA.APLI_IMPOR,
0 AS SALDO_PENDIENTE,
DATEDIFF(DD, COM.FECHA, CE.FECHA) AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
     ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= 7 THEN 'A - DE 0 A 7 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 7 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=15 THEN 'B - DE 8 A
15 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 15 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=21 THEN 'C - DE 16 A
21 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 21 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=28 THEN 'D - DE 22 A
28 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 28 AND DATEDIFF(DD, COM.FECHA, CE.FECHA) <=35 THEN 'E - DE 29 A
35 DIAS'
      WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) > 35 THEN 'F - MAS DE 35 DIAS'
END AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
CASE WHEN DATEDIFF(DD, COM.FECHA, CE.FECHA) <= DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO ) THEN 'OK' ELSE
'NO' END AS AUDITORIA,
COM.OBSER

FROM factu_full_central..COMPROMO_API CA
    INNER JOIN factu_full_central..COMP C
    ON CA.TIPO = C.COD_COMP
    INNER JOIN factu_full_central..COMP_EMITIDOS CE
    ON CA.TIPO = CE.TIPO
    AND CA.NUM = CE.NUM
    INNER JOIN factu_full_central..COMP_EMITIDOS AS COM
    ON COM.TIPO = CA.APLI_TIPO
    AND COM.NUM = CA.APLI_NUM
    LEFT JOIN factu_full_central..CLIENTES AS CLI

```

```

ON COM.CLIENTE = CLI.NUM_CLIENTE

WHERE CLI.AGRU_3 = 'FULL'
AND CONCAT (CA.API_NUM,'-',RIGHT (CONCAT ('0000000000',CA.API_NUM ),8)) IN
(SELECT DISTINCT CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO
FROM factu_full_central..COMP_EMITIDOS AS COM
WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO = 'PEN'
AND YEAR(COM.FECHA) >= year(dateadd(yy,-1, getdate())))

UNION ALL
-----
----- GRUPO C.1 - FACTURAS CON ALGUN PAGO EFECTUADO - SOLO TOMO EL SALDO DE LAS FACTURAS
-----
SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO AS ESTADO,
CONCAT(COM.ESTADO,'-PAR') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
      ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
      ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
      ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO)
      ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,
'PENDIENTE' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 7 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 15 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) > 21 AND DATEDIFF(DD,GETDATE(),COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'

```

```

WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28 THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'F%'
AND COM.ESTADO IN ('PEN')
AND COM.SALDO <> 0
AND COM.SALDO <> COM.TOTAL
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'

```

```
UNION ALL
```

```
----- GRUPO D - ANTICIPOS SIN APLICAR
```

```

SELECT 'FULL' AS EMPRESA,
CONCAT (COM.TIPO,'-',RIGHT (CONCAT ('0000000000',COM.NUM),8)) AS FACTURA_NRO,
COM.ESTADO,
CONCAT(COM.ESTADO,'-ANT') AS ESTADO_CLAVE,
COM.FECHA AS FECHA_FACTURA,
COM.FECHA_PAGO AS FECHA_VENCIMIENTO,
COM.FECHA_PAGO AS FECHA_COBRANZA,
DATEPART(DW,COM.FECHA) AS DIA_VENCIMIENTO,
DATEDIFF (DD, COM.FECHA , COM.FECHA_PAGO) AS DIAS_PAGO,
DATEPART (WEEK,COM.FECHA) AS SEMANA_EMISION,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_VENCIMIENTO,
DATEPART (WEEK,COM.FECHA_PAGO) AS SEMANA_PAGO_REAL,
CASE WHEN LEN(MONTH(COM.FECHA)) = 1 THEN CONCAT(YEAR(COM.FECHA),'-0',MONTH(COM.FECHA))
     ELSE CONCAT(YEAR(COM.FECHA),'-',MONTH(COM.FECHA)) END AS YYYYMM_EMISION,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_VENCIMIENTO,
CASE WHEN LEN(MONTH(COM.FECHA_PAGO)) = 1 THEN CONCAT(YEAR(COM.FECHA_PAGO),'-0',MONTH(COM.FECHA_PAGO))
     ELSE CONCAT(YEAR(COM.FECHA_PAGO),'-',MONTH(COM.FECHA_PAGO)) END AS YYYYMM_PAGO_REAL,
0 AS API_IMPOR,
COM.SALDO AS SALDO_PENDIENTE,
0 AS DIAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 THEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO)
     ELSE 0 END AS DIAS_AL_VENCIMIENTO,
COM.CLIENTE,
CLI.NOM_FANTASIA,

```

```

'SIN VENCER' AS GAP_COBRANZAS,
CASE WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) < 0 THEN 'A - VENCIDO'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) >= 0 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=7 THEN
'B - DE 1 A 7 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 7 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=15 THEN
'C - DE 8 A 15 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 15 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=21 THEN
'D - DE 16 A 21 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 21 AND DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) <=28 THEN
'E - DE 22 A 28 DIAS'
      WHEN DATEDIFF(DD, GETDATE(), COM.FECHA_PAGO) > 28  THEN 'F - MAS DE 35 DIAS'
END AS GAP_PENDIENTE,
COM.TOTAL AS TOTAL_FACTURA,
'OK' AS AUDITORIA,
COM.OBSER

```

```

FROM factu_full_central..COMP_EMITIDOS AS COM
INNER JOIN factu_full_central..CLIENTES AS CLI
ON CLI.NUM_CLIENTE = COM.CLIENTE

```

```

WHERE COM.TIPO LIKE 'A%'
AND COM.ESTADO NOT IN ('CAN')
AND COM.SALDO <> 0
AND YEAR(COM.FECHA) >= year(dateadd(yy, -1, getdate()))
AND CLI.AGRU_3 = 'FULL'
AND COM.SALDO = COM.TOTAL

```

Rotacion por producto

La idea es obtener una tabla que nos de por categoría califique al producto como Alta, Media y Baja rotacion.

- ventas de los últimos 6 meses
- por categoría de producto se obtiene esta tabla
 - a. Total de unidades vendidas por los últimos 6 meses de productos Cxxx y los combos que incluyan el mismo Cxxx
 - b. Excluir la venta mayorista, agrupacion 3 es "Venta canal mayorista" codigo MAYOR
 - c. Cantidad de SKU's de la misma categoría
 - d. De este numero se calcula 20% para arriba y para abajo

Ejemplo -> Total de ventas 91394 / 6 meses ☐ consumo mensual 15232

Ese numero se divide por la cantidad de skus en el mismo periodo da 236

La cuenta es 64.54

☐ este numero se multiplica por 1.20 ☐ alta ☐ 78 (redondeo para arriba)

☐ este numero se multiplica por 0.80 ☐ baja ☐ 51 (redondeo para abajo)

todo lo que este dentro de 51 y 78 es media

Se tiene que grabar todos los ultimos dias del mes junto al SP que genera la tabla pbi_stock_depositos con el SP XXXXXXXXXXXXXXXX [@Augusto](#) buscalo para sumarle el SP para que tambien grabe la tabla pbi_rotacion_historica

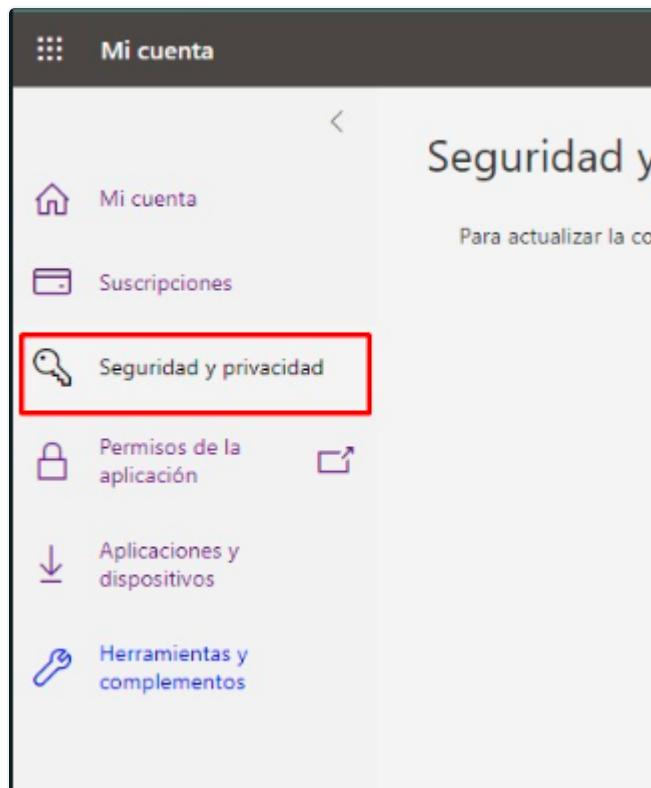
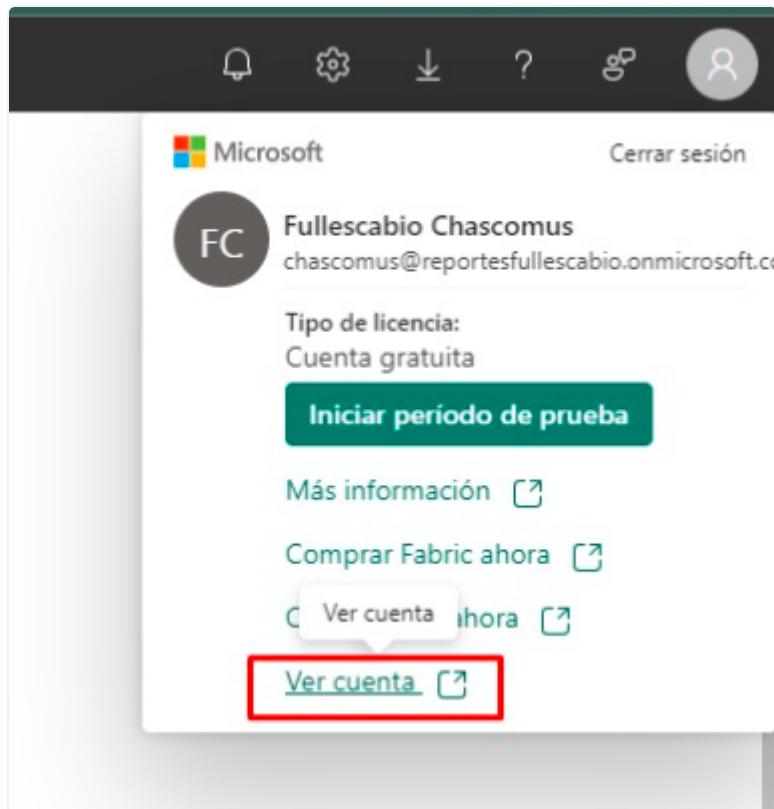
/

[@Augusto](#) armar 2023 completo, revisar con Lu y con el OK hablar con Adrian para que deje de calcularlo y use la tabla

[@Augusto](#) borrar PBI stock 2.0

•

Agregar permisos Authenticator



Seguridad y privacidad

Para actualizar la contraseña, las preferencias de contacto o para ver la declaración de privacidad de su organización, vaya al nuevo [Portal de Mi cuenta](#)

Mis inicios de sesión ▾

Información general

Información de seguri...

Dispositivos

Contraseña

Organizaciones

Configuración y privaci...

Actividad reciente

Información de seguridad

Estos son los métodos que usa para iniciar sesión en su cuenta o restablecer su contraseña.

Método de inicio de sesión predeterminado: Microsoft Authenticator: notificación [Cambiar](#)

[+ Agregar método de inicio de sesión](#)

...	Contraseña	Última actualización: hace 6 meses
👤	Microsoft Authenticator Insertar autenticación multifactor (MFA)	SM-M127F
👤	Microsoft Authenticator Insertar autenticación multifactor (MFA)	SM-G990E

¿Ha perdido el dispositivo? [Cerrar todas las sesiones](#)

PBI - Logistica Pedidos

PBI - Segmentador ADLC

Vistas

full_powerbi_kpi_empresa_articulos
full_powerbi_KPI_Sucursales_Sucursales
full_powerbi_KPI_Sucursales_Subcategorias
full_powerbi_KPI_Sucursales_Productos
full_powerbi_KPI_Sucursales_Componentes
full_powerbi_KPI_Sucursales_Marcas
full_powerbi_KPI_Sucursales_Categorias
full_powerbi_KPI_Sucursales_Comprobantes
full_powerbi_KPI_Sucursales_Clientes

Tabla

pbisucursales_canales
pbisucursales_tipo_producto
pbiventas_locales

PBI Ventas sucursales

Cuando se ingresa una nueva sucursal. estos son los pasos a seguir

1 - Modificar la tabla empresa en la base de locales para que el agrupacion_sucursal este alineado a las zonas establecidas

```
use factu_full_locales  
--select * from EMPRESAs where COMP_GRUPO = 'SUA',  
  
update EMPRESAS set AGRUPACION_SUCURSAL = 'CABA' where COMP_GRUPO = 'C02'
```

Esto soluciona los blancos en la vista □ Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-6544>)

Amigos de la casa

Se usa esta vista para liquidar las comisiones de amigos de la casa.

Campo ADLC_OBSERVACIONES

```
CASE WHEN COM.CLIENTE < 100 THEN 'APP'  
      WHEN (ISNULL(year(com.fecha_sincro_thalamus),0) = 0  
            AND TIPO_COMP.TIPO <> 'A'  
            AND COM.FULL_VERSION_PDV is null) THEN 'NO-PDV VIEJO'  
  
      WHEN COM.CUIT_COMPRO = '11111111' THEN 'NO-CONS FINAL'  
      WHEN COM.CLIENTE = 104 THEN 'NO-CONS FINAL'  
      WHEN TIPO_COMP.TIPO = 'A' THEN 'FC A'  
  
ELSE 'SI' END AS ADLC_OBSERVACIONES,
```

Campo ADLC_SI o NO

Es lo mismo para este campo, pero usamos el SI o NO para calcular la comision segun el % de la penetracion del mes anterior

```
CASE WHEN COM.CLIENTE < 100 THEN 'APP'  
      WHEN (ISNULL(year(com.fecha_sincro_thalamus),0) = 0  
            AND TIPO_COMP.TIPO <> 'A'  
            AND COM.FULL_VERSION_PDV is null) THEN 'NO'  
  
      WHEN COM.CUIT_COMPRO = '11111111' THEN 'NO'  
      WHEN COM.CLIENTE = 104 THEN 'NO'  
      WHEN TIPO_COMP.TIPO = 'A' THEN 'FC A'  
ELSE 'SI' END AS ADLC
```

Campo ES.DTO

Determinación de si es descuento o no, los productos siempre deben comenzar con DTO- para que sean reintegrados por esta liquidación

```
CASE WHEN SUBSTRING(ITM.ARTICULO,1,4) = 'DTO-' THEN 'SI'  
ELSE 'NO' END AS ES_DTO
```

Controles

- SI "ADLC Observaciones" es APP, el cliente siempre debe ser menor a 100 y el campo ES_DTO siempre debe ser "NO"
- SI "ADLC Observaciones" es FC A, todos los comprobantes deben ser FA A y el campo ES_DTO debe ser siempre "NO"
- SI "ADLC Observaciones" es NO-CONS FINAL, son los casos en donde el cliente es 100 al 104, el cuit es "11111111" y no tiene numero de documento, El campo ES_DTO siempre debe ser "NO"
- SI "ADLC Observaciones" es NO-PDV VIEJO, son los casos en donde la fecha de sincro con thalamus es nula y la versión del PDV es nula tambien, el campo ES_DTO siempre debe ser "NO" □ en caso de que sea "SI" hay que cambiar manualmente el campo "ADLC" a "SI" y poner en ADLC_Observaciones la leyenda "error de sincronizacion con thalamus".
Esta venta debe si o si enviarse a Thalamus por FTP nocturno
- SI "ADLC Observaciones" es "SI" el campo ADLC siempre debe decir "SI" y el campo ES_DTO puede ser SI o NO, segun el tipo de producto.

FTP Thalamus

Extracción usada para el envío de ventas y artículos por FTP a Thalamus

V - thalamus_reporte_ventas

- SQL

```
select
CONCAT('1:',CLI.CUIT) collate database_default as credentialprincipal
,ITM.ARTICULO collate database_default as EAN
,CONVERT(NUMERIC(10),ABS(ITM.CANT)) as Quantity
,CASE WHEN left(ITM.ARTICULO,3) IN ('DTO','RED') THEN CONVERT(NUMERIC(10,2),ITM.CANT * -
ITM.PRECIO_NETO * 1.21)
when ITM.PORCEN_IVA = 0 then CONVERT(NUMERIC(10,2),ABS(ITM.CANT * ITM.PRECIO_NETO))
WHEN left(ITM.TIPO_FACT,1) = 'C' THEN CONVERT(NUMERIC(10,2),ITM.CANT * - ITM.PRECIO_NETO * 1.21)
ELSE CONVERT(NUMERIC(10,2),ABS(ITM.CANT * convert(numeric(10,3),ITM.PRECIO_NETO * 1.21))) END as
Amount
,CONVERT(VARCHAR,COM.FECHA,23) collate database_default as 'Sale Date'
,' as Branch
,' as 'Payment Method'--VAL.DESCRIP_VALOR as 'Payment Method'
,' as Currency
,' as ExternalVendorId
,' as ExternalVendorDescription
,' as VendorId
,CONCAT(COM.TIPO,'-',COM.NUM) collate database_default as TransactionId
,trim(REPLACE(REPLACE(KIT.Location_ID,CHAR(13),''),CHAR(10),'')) collate database_default as
PhysicalLocationId
,' as 'Description 1'
,' as 'Description 2'
,' as 'Description 3'
,' as 'Description 4'
,' as 'Description 5'
,' as 'Description 6'
,' as 'Description 7'
,' as 'Description 8'
,' as 'Description 9'
,' as 'Description 10'
from factu_full_locales.dbo.COMP_EMITIDOS AS COM
INNER JOIN factu_full_locales.dbo.RENG_FAC AS ITM ON (COM.TIPO=ITM.TIPO_FACT AND
COM.NUM=ITM.NUM_FACT )
INNER JOIN factu_full_central.dbo.KIKKER_TIENDA AS KIT ON ITM.DEP0=KIT.DEP0
INNER JOIN factu_full_locales.dbo.CLIENTES AS CLI ON COM.CLIENTE=CLI.NUM_CLIENTE
WHERE COM.FECHA = (select CONVERT(varchar,getdate() -1,23))
and LEFT(tipo,1) in ('F','C')
AND ITM.COMPO_ITEM IS NULL
and RIGHT(isnull(obser_sincro_thalamus,'XX'),2) not in ('OK','IT')
and cli.IVA = 4
```

```

union all

select
CONCAT('1:', CLI.CUIT) as credentialprincipal
, ITM.ARTICULO as EAN
, CONVERT(NUMERIC(10), ABS(ITM.CANT)) as Quantity
, CASE WHEN left(ITM.ARTICULO,3) IN ('DTO','RED') THEN CONVERT(NUMERIC(10,2), ITM.CANT * -
ITM.PRECIO_NETO * 1.21)
when ITM.PORCEN_IVA = 0 then CONVERT(NUMERIC(10,2),ABS(ITM.CANT * ITM.PRECIO_NETO))
WHEN left(ITM.TIPO_FACT,1) = 'C' THEN CONVERT(NUMERIC(10,2),ITM.CANT * - ITM.PRECIO_NETO * 1.21)
ELSE CONVERT(NUMERIC(10,2),ABS(ITM.CANT * convert(numeric(10,3),ITM.PRECIO_NETO * 1.21)))
END as Amount
, CONVERT(VARCHAR, COM.FECHA, 23) as 'Sale Date'
,' as Branch
,' as 'Payment Method'--VAL.DESCRIP_VALOR as 'Payment Method'
,' as Currency
,' as ExternalVendorId
,' as ExternalVendorDescription
,' as VendorId
, CONCAT(COM.TIPO, ' - ', COM.NUM) as TransactionId
, trim(REPLACE(REPLACE(KIT.Location_ID,CHAR(13), '' ),CHAR(10), '' )) as PhysicalLocationId
,' as 'Description 1'
,' as 'Description 2'
,' as 'Description 3'
,' as 'Description 4'
,' as 'Description 5'
,' as 'Description 6'
,' as 'Description 7'
,' as 'Description 8'
,' as 'Description 9'
,' as 'Description 10'
from factu_full_bahia_blanca.dbo.COMP_EMITIDOS AS COM
INNER JOIN factu_full_bahia_blanca.dbo.RENG_FAC AS ITM ON (COM.TIPO=ITM.TIPO_FACT AND
COM.NUM=ITM.NUM_FACT)
INNER JOIN factu_full_central.dbo.KIKKER_TIENDA AS KIT ON ITM.DEPO=KIT.DEPO collate
database_default
INNER JOIN factu_full_bahia_blanca.dbo.CLIENTES AS CLI ON COM.CLIENTE=CLI.NUM_CLIENTE
WHERE COM.FECHA = (select CONVERT(varchar,getdate() -1,23))
and LEFT(tipo,1) in ('F','C')
AND ITM.COMPO_ITEM IS NULL
and RIGHT(isnull(obser_sincro_thalamus,'XX'),2) not in ('OK','IT')
and cli.IVA = 4

```

Campos que se envían

SQL

Comentario

CONCAT('1:',CLI.CUIT) collate database_default as credentialprincipal	1; + numero de documento
,ITM.ARTICULO collate database_default as EAN	Codigo de articulo
,CONVERT(NUMERIC(10),ABS(ITM.CANT)) as Quantity	Cantidad
,CASE WHEN left(ITM.ARTICULO,3) IN ('DTO','RED') THEN CONVERT(NUMERIC(10,2),ITM.CANT * - ITM.PRECIO_NETO * 1.21)	Monto - Los articulos de desuento y redondeo tienen valor en negativo + IVA
when ITM.PORCEN_IVA = 0 then CONVERT(NUMERIC(10,2),ABS(ITM.CANT * ITM.PRECIO_NETO))	Monto - Cuando el articulo no tiene iva se pasa el monto * cantidad
WHEN left(ITM.TIPO_FACT,1) = 'C' THEN CONVERT(NUMERIC(10,2),ITM.CANT * - ITM.PRECIO_NETO * 1.21)	Monto - Los articulos de una NC se pasan como negativo + IVA
ELSE CONVERT(NUMERIC(10,2),ABS(ITM.CANT * convert(numeric(10,3),ITM.PRECIO_NETO * 1.21))) END as Amount	Monto - Se envia el precio + IVA
,CONVERT(VARCHAR,COM.FECHA,23) collate database_default as 'Sale Date'	Fecha del comprobante
," as Branch," as 'Payment Method'," as Currency," as ExternalVendorId," as ExternalVendorDescription," as VendorId	Valores que se envian en blanco
,CONCAT(COM.TIPO,'-',COM.NUM) collate database_default as TransactionId	Tipo y numero de comprobante
,trim(REPLACE(REPLACE(KIT.Location_ID,CHAR(13),"'),CHAR(10),"')) collate database_default as PhysicalLocationId	Numero de ID dentro de Thalamus
," as 'Description 1',' as 'Description 2',' as 'Description 3',' as 'Description 4',' as 'Description 5',' as 'Description 6',' as 'Description 7',' as 'Description 8',' as 'Description 9',' as 'Description 10'	Valores que se envian en blanco

Los comprobantes se filtran de la siguiente manera

- Con fecha del día anterior
- De tipo Factura y Note de Crédito
- No se envían componentes
- Donde el campo de respuesta de thalamus no es OK
- Los clientes son Consumidor Final

```
WHERE COM.FECHA = (select CONVERT(varchar,getdate() -1,23))
and LEFT(tipo,1) in ('F','C')
AND ITM.COMPO_ITEM IS NULL
and RIGHT(isnull(obser_sincro_thalamus,'XX'),2) not in ('OK','IT')
and cli.IVA = 4
```

V - thalamus_reporte_articulos

.

SQL

```
select
art.cod_articulo as 'Product Code',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') as 'ProductName',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') as 'ProductDescription',
'VERDADERO'

'Product Active',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') as 'Sku Name',
art.cod_articulo as 'EAN',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') as 'SkuDescription',
'VERDADERO' 'Sku Active',
'1' as 'Number of Units',
'1' as 'Selling Quantities',
'' as 'Description 1',
'' as 'Description 2',
'' as 'Description 3',
'' as 'Description 4',
'' as 'Description 5',
'' as 'Description 6',
'' as 'Description 7',
'' as 'Description 8',
'' as 'Description 9',
'' as 'Description 10',
'' as 'Description 11',
'' as 'Description 12',
'' as 'Description 13',
'' as 'Description 14',
'' as 'Description 15',
'' as 'Description 16',
'' as 'Description 17',
'' as 'Description 18',
'' as 'Description 19',
'' as 'Description 20',
'' as 'Description 21',
'' as 'Description 22',
'' as 'Description 23',
'' as 'Description 24',
'' as 'Description 25',
'' as 'Description 26',
```

```

    '' as 'Description 27',
    '' as 'Description 28',
    '' as 'Description 29',
    '' as 'Description 30',
    '' as 'Description 31',
    '' as 'Description 32',
    '' as 'Description 33',
    '' as 'Description 34',
    '' as 'Description 35',
    '' as 'Description 36',
    '' as 'Description 37',
    '' as 'Description 38',
    '' as 'Description 39',
    '' as 'Description 40',
    '' as 'Relationships Ean',
translate(translate(translate(translate(translate(replace(replace(AGR.DESCRIPT_AGRU,
CHAR(13), ''), CHAR(10), '')
    , ' ', '_')
    , 'á', 'a')
    , 'é', 'e')
    , 'í', 'i')
    , 'ó', 'o')
    , 'ú', 'u') as 'Brand Code',
replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), ''), CHAR(10), '') as 'Brand Name',
replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), ''), CHAR(10), '') as 'Brand Description',
'' as 'Brand Description1',
'' as 'Brand Description2',
'' as 'Brand Description3',
'' as 'Brand Description4',
'' as 'Brand Description5',
'VERDADERO' as 'Brand Active',
translate(translate(translate(translate(translate(replace(replace(AGR.DESCRIPT_AGRU,
CHAR(13), ''), CHAR(10), '')
    , ' ', '_')
    , 'á', 'a')
    , 'é', 'e')
    , 'í', 'i')
    , 'ó', 'o')
    , 'ú', 'u') as 'Brand Family Code',
replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), ''), CHAR(10), '') as 'Brand Family Name',
replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), ''), CHAR(10), '') as 'Brand Family Description',
'' as 'Brand Family Description1',
'' as 'Brand Family Description2',
'' as 'Brand Family Description3',
'' as 'Brand Family Description4',
'' as 'Brand Family Description5',
'' as 'Brand Family Use Thalamus',
'VERDADERO' as 'Age Restriction',
'VERDADERO' as 'Brand Family Active',
'fullescabio' as 'Company Code',

```

```

    '' as 'Company Name',
    '' as 'Company Active',
    '' as 'Company Use Thalamus',
    'fullescabio' as 'Market Code',
    '' as 'Market Name',
    '' as 'Market Description',
    '' as 'Market Active'

from factu_full_locales.dbo.ARTICULOS as art
INNER JOIN factu_full_locales.dbo.AGRUPACIONES AS AGR ON ART.AGRU_1=AGR.CODI_AGRU and agr.NUM_AGRU
= 1
WHERE art.FECHA_ALTA >= (SELECT CONVERT(varchar, GETDATE() -1,23))
and agr.CODI_AGRU in ('1008',
'1009',
'1031',
'1012',
'1028',
'1024',
'1034',
'1103',
'1032',
'1015',
'1002',
'1007',
'1029',
'1001',
'1033',
'1025',
'1006',
'1026',
'1021',
'1039',
'1016',
'1019',
'1027',
'1013',
'1018',
'1003',
'1022',
'1020',
'1030',
'1005',
'1004',
'1023',
'1017'
)

UNION

select
art.cod_articulo COLLATE DATABASE_DEFAULT as 'Product Code',

```

```
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') COLLATE
DATABASE_DEFAULT as 'Product Name',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') COLLATE
DATABASE_DEFAULT as 'Product Description',
'VERDADERO' 'Product Active',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') COLLATE
DATABASE_DEFAULT as 'Sku Name',
art.cod_articulo COLLATE DATABASE_DEFAULT as 'EAN',
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ''), CHAR(10), ''), ',', '.') COLLATE
DATABASE_DEFAULT as 'Sku Description',
'VERDADERO' 'Sku Active',
'1' as 'Number of Units',
'1' as 'Selling Quantities',
'' as 'Description 1',
'' as 'Description 2',
'' as 'Description 3',
'' as 'Description 4',
'' as 'Description 5',
'' as 'Description 6',
'' as 'Description 7',
'' as 'Description 8',
'' as 'Description 9',
'' as 'Description 10',
'' as 'Description 11',
'' as 'Description 12',
'' as 'Description 13',
'' as 'Description 14',
'' as 'Description 15',
'' as 'Description 16',
'' as 'Description 17',
'' as 'Description 18',
'' as 'Description 19',
'' as 'Description 20',
'' as 'Description 21',
'' as 'Description 22',
'' as 'Description 23',
'' as 'Description 24',
'' as 'Description 25',
'' as 'Description 26',
'' as 'Description 27',
'' as 'Description 28',
'' as 'Description 29',
'' as 'Description 30',
'' as 'Description 31',
'' as 'Description 32',
'' as 'Description 33',
'' as 'Description 34',
'' as 'Description 35',
'' as 'Description 36',
'' as 'Description 37',
```

```

'' as 'Description 38',
'' as 'Description 39',
'' as 'Description 40',
'' as 'Relationships Ean',
translate(transl
ate(transl
ate(transl
ate(transl
ate(replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '')
, ' ', '_')
, 'á', 'a')
, 'é', 'e')
, 'í', 'i')
, 'ó', 'o')
, 'ú', 'u') COLLATE DATABASE_DEFAULT as 'Brand Code',
replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '') COLLATE DATABASE_DEFAULT as 'Brand Name',
replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '') COLLATE DATABASE_DEFAULT as 'Brand Description',
'' as 'Brand Description1',
'' as 'Brand Description2',
'' as 'Brand Description3',
'' as 'Brand Description4',
'' as 'Brand Description5',
'VERDADERO' as 'Brand Active',
translate(transl
ate(transl
ate(transl
ate(transl
ate(replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '')
, ' ', '_')
, 'á', 'a')
, 'é', 'e')
, 'í', 'i')
, 'ó', 'o')
, 'ú', 'u') COLLATE DATABASE_DEFAULT as 'Brand Family Code',
replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '') COLLATE DATABASE_DEFAULT as 'Brand Family Name',
replace(replace(AGR.DESCRIPT
AGR, CHAR(13), ''), CHAR(10), '') COLLATE DATABASE_DEFAULT as 'Brand Family Description',
'' as 'Brand Family Description1',
'' as 'Brand Family Description2',
'' as 'Brand Family Description3',
'' as 'Brand Family Description4',
'' as 'Brand Family Description5',
'' as 'Brand Family Use Thalamus',
'VERDADERO' as 'Age Restriction',
'VERDADERO' as 'Brand Family Active',
'fullescabio' as 'Company Code',
'' as 'Company Name',
'' as 'Company Active',
'' as 'Company Use Thalamus',
'fullescabio' as 'Market Code',
'' as 'Market Name',
'' as 'Market Description',
'' as 'Market Active'

```

```
from FACTU_FULL_BAHIA_BLANCA.DBO.ARTICULOS as art
INNER JOIN FACTU_FULL_BAHIA_BLANCA.DBO.AGRUPACIONES AS AGR ON ART.AGRU_1=AGR.CODI_AGRU and
agr.NUM_AGRU=1
WHERE art.FECHA_ALTA >= (SELECT CONVERT(varchar,GETDATE() -1,23))
and agr.CODI_AGRU in ('1008',
'1009',
'1031',
'1012',
'1028',
'1024',
'1034',
'1103',
'1032',
'1015',
'1002',
'1007',
'1029',
'1001',
'1033',
'1025',
'1006',
'1026',
'1021',
'1039',
'1016',
'1019',
'1027',
'1013',
'1018',
'1003',
'1022',
'1020',
'1030',
'1005',
'1004',
'1023',
'1017')
```

Seleccion de Campos

SQL

art.cod_articulo as 'Product Code',

translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), "'), CHAR(10), "'), '','')) as 'Product Name',

```
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ""), CHAR(10), ""),'','','.') as 'Product Description',
```

```
'VERDADERO' 'Product Active',
```

```
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ""), CHAR(10), ""),'','','.') as 'Sku Name',
```

```
art.cod_articulo as 'EAN',
```

```
translate(replace(replace(art.DESCRIP_ARTI, CHAR(13), ""), CHAR(10), ""),'','','.') as 'Sku Description',
```

'VERDADERO' 'Sku Active','1' as 'Number of Units','1' as 'Selling Quantities',

" as 'Description 1'," as 'Description 2'," as 'Description 3'," as 'Description 4'," as 'Description 5'," as 'Description 6'," as 'Description 7'," as 'Description 8'," as 'Description 9'," as 'Description 10'," as 'Description 11'," as 'Description 12'," as 'Description 13'," as 'Description 14'," as 'Description 15'," as 'Description 16'," as 'Description 17'," as 'Description 18'," as 'Description 19'," as 'Description 20'," as 'Description 21'," as 'Description 22'," as 'Description 23'," as 'Description 24'," as 'Description 25'," as 'Description 26'," as 'Description 27'," as 'Description 28'," as 'Description 29'," as 'Description 30'," as 'Description 31'," as 'Description 32'," as 'Description 33'," as 'Description 34'," as 'Description 35'," as 'Description 36'," as 'Description 37'," as 'Description 38'," as 'Description 39'," as 'Description 40'," as 'Relationships Ean',

translate(translate(translate(translate(translate(translate(replace(replace(AGR.DESCRIP_AGRU,
CHAR(13), ")), CHAR(10), ") , ' ','_') , 'á','a') , 'é','e') , 'í','i') , 'ó','o')
, 'ú','u') as 'Brand Code',

replace(replace(AGR.DESCRIP_AGRU, CHAR(13), ")), CHAR(10), ") as 'Brand
Name', replace(replace(AGR.DESCRIP_AGRU, CHAR(13), ")), CHAR(10), ") as 'Brand Description',

" as 'Brand Description1'," as 'Brand Description2'," as 'Brand Description3'," as 'Brand
Description4'," as 'Brand Description5',

'VERDADERO' as 'Brand Active',

translate(translate(translate(translate(translate(replace(replace(AGR.DESCRIPT_AGRU,
CHAR(13), ")), CHAR(10), ") , ''_') , 'á','a') , 'é','e') , 'í','i') , 'ó','o')
, 'ú','u') as 'Brand Family Code',

replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), "), CHAR(10), ") as 'Brand Family
Name', replace(replace(AGR.DESCRIPT_AGRU, CHAR(13), "), CHAR(10), ") as 'Brand Family
Description',

" as 'Brand Family Description1'," as 'Brand Family Description2'," as 'Brand Family
Description3'," as 'Brand Family Description4'," as 'Brand Family Description5'," as 'Brand
Family Use Thalamus','VERDADERO' as 'Age Restriction','VERDADERO' as 'Brand Family
Active','fullescabio' as 'Company Code'," as 'Company Name'," as 'Company Active'," as
'Company Use Thalamus','fullescabio' as 'Market Code'," as 'Market Name'," as 'Market
Description'," as 'Market Active'

Se envían los artículos que se dieron de alta el día anterior y que correspondan a una de estas agrupaciones

- - Agrupaciones
 - 1001 Accesorios
 - 1002 Aguardientes
 - 1003 Aperitivos
 - 1004 Cervezas Lata

- 1005 Cervezas Litro
- 1006 Cervezas Porron
- 1007 Champagnes
- 1008 Combos
- 1009 Delicatessen
- 1012 Espumantes
- 1013 Gin
- 1015 Hielo
- 1016 Insumos
- 1017 Kits Mercado Libre
- 1018 Licores
- 1019 Otros
- 1020 Productos Promocionales
- 1021 Recargo Costo de envio
- 1022 Regalería
- 1023 Ron
- 1024 Sidra
- 1025 Sin Alcohol
- 1026 Tequilas
- 1027 Vinos
- 1028 Vodka
- 1029 Whiskies
- 1030 Combos
- 1031 Snacks
- 1032 Hard Seltzer
- 1033 Coolers
- 1034 Cristaleria
- 1039 RP
- 1103 FullEscabio

```
WHERE art.FECHA_ALTA >= (SELECT CONVERT(varchar, GETDATE() -1, 23))
and agr.CODI_AGRU in ('1008',
'1009',
'1031',
'1012',
'1028',
'1024',
'1034',
'1103',
'1032',
'1015',
'1002',
'1007',
'1029',
'1001',
'1033',
'1025',
'1006',
'1026',
'1021',
'1039',
'1016',
'1019',
'1027',
'1013',
'1018',
'1003',
'1022',
'1020',
'1030',
'1005',
'1004',
'1023',
'1017'
)
```

Dinamicas en RP

DBA

DBA_CONSULTA_STOCK

```
SELECT
DEP.DEPO,
'DBA Centro' as DEPOSITO,
ART.COD_ARTICULO,
ART.DESCRIP_ARTI,
ART.CANT_STOCK ART_CANT_STOCK,
DEP.CANT_STOCK DEP_CANT_STOCK,
CASE WHEN RES.RESERVADA IS NULL THEN DEP.CANT_STOCK
     ELSE DEP.CANT_STOCK- RES.RESERVADA END AS STOCK_DISPONBIBLE
FROM factu_dba.dbo.ARTICULOS AS ART
LEFT JOIN factu_dba.dbo.ARTICULOS_STOCK_DEPO AS DEP
ON ART.COD_ARTICULO = DEP.COD_ARTICULO
LEFT JOIN (SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA-- dbo.PEDI_ITEMS
          FROM factu_dba.dbo.PEDI_ITEMS
          WHERE RESER = 'S' AND CANT_PEND <> 0 GROUP BY ARTICULO, DEPO_RESER ) AS RES
ON DEP.COD_ARTICULO = RES.ARTICULO
AND DEP.DEPO = RES.DEPO_RESER
WHERE ART.ACTIVO = 'S'
AND DEP.DEPO = 'DEP'
UNION ALL
SELECT
DEP.DEPO,
'DBA Norte' as DEPOSITO,
ART.COD_ARTICULO,
ART.DESCRIP_ARTI,
ART.CANT_STOCK ART_CANT_STOCK,
DEP.CANT_STOCK DEP_CANT_STOCK,
CASE WHEN RES.RESERVADA IS NULL THEN DEP.CANT_STOCK
     ELSE DEP.CANT_STOCK- RES.RESERVADA END AS STOCK_DISPONBIBLE
FROM factu_dba_norte.dbo.ARTICULOS AS ART
LEFT JOIN factu_dba_norte.dbo.ARTICULOS_STOCK_DEPO AS DEP
ON ART.COD_ARTICULO = DEP.COD_ARTICULO
LEFT JOIN (SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA-- dbo.PEDI_ITEMS
          FROM factu_dba_norte.dbo.PEDI_ITEMS
          WHERE RESER = 'S' AND CANT_PEND <> 0 GROUP BY ARTICULO, DEPO_RESER ) AS RES
ON DEP.COD_ARTICULO = RES.ARTICULO
AND DEP.DEPO = RES.DEPO_RESER
WHERE ART.ACTIVO = 'S'
AND DEP.DEPO = 'DEP'
```

Vista creada para el archivo de consulta de stock para los dos dba

/co

Cambios de vistas

Codigo original de las vistas de la base DBA NORTE antes de pisarlos con el codigo de la base de DBA Centro

- **DBA_VENDE_PBI**

```
ALTER VIEW [dbo].[DBA_VENDE_PBI] AS
SELECT CODI_VENDE ,NOMBRE ,INACTIVO ,
CASE WHEN CODI_VENDE IN ('7','10') THEN 'N' ELSE 'S' END AS CONSIDERAR,
CASE WHEN CODI_VENDE IN ('9','37','40','51','60','65','61','62','63','52') THEN 'PLATAFORMAS'
ELSE 'ON' END AS TIPO_CANAL,
CASE WHEN CODI_VENDE = '9' THEN 'Ecommerce'
WHEN CODI_VENDE = '37' THEN 'Wabi'
WHEN CODI_VENDE = '40' THEN 'Tecnogo'
WHEN CODI_VENDE = '51' THEN 'Tecnogo ML'
WHEN CODI_VENDE = '60' THEN 'Tecnogo'
WHEN CODI_VENDE = '65' THEN 'Tecnogo Shopee'
WHEN CODI_VENDE = '61' THEN 'Tecnogo'
WHEN CODI_VENDE = '62' THEN 'Tecnogo Shopify'
WHEN CODI_VENDE = '63' THEN 'ICBC Mall'
WHEN CODI_VENDE = '52' THEN 'Tecnogo ML'
END AS TIPO_PLATAFORMA,
'dba' AS BASE
FROM VENDEDORES v;
```

- **DBA_VENTAS_PROVEEDORES_VISTA**

```
ALTER view [dbo].[dba_ventas_proveedores_vista]
as
-- select * from dba_ventas_proveedores_vista where fecha_datetime between '20210301' and
'20210301'

SELECT

-- CAMBIO DEL CAMPO CLIENTE SEGUN LA AGRUPACION 3
COM.CLIENTE,

-- CODIGO DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL

COM.CLIENTE as COD_CLIENTE_ORIGINAL,

-- CAMBIO DE NOMBRE DE CLIENTE SEGUN LA AGRUPACION 3
```

```
CLI.NOM_FANTASIA ,  
-- NOMBRE DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL  
  
CLI.NOM_FANTASIA AS NOMBRE_CLIENTE_ORIGINAL,  
  
-- ID TRIBUTARIA SEGUIN AGRUPACION 3  
  
CASE WHEN CLI.IVA = '4' THEN 'D' else 'C' end TIPO_ID_TRIBUTARIA,  
  
-- CUIT DEL CLIENTE O CLIENTE FINAL SEGUN AGRUPACION 3  
  
CLI.CUIT ID_TRIBUTARIA,  
  
-- RAZON SOCIAL DEL CLIENTE SEGUIN AGRUPACION 3  
  
CLI.RAZON RAZON_SOCIAL_CLIENTE,  
  
-- RAZON SOCIAL ORIGINAL -> NO PASAR NUNCA, ES CONTROL  
  
CLI.RAZON AS RAZON_SOCIAL_ORIGINAL,  
  
-- PROVINCIA SEGUN AGRUPACION 3  
CLI.PROVINCIA PROVINCIA,  
  
-- LOCALIDAD SEGUN AGRUPACION 3  
CLI.LOCALIDAD,  
  
-- DOMICILIO SEGUN AGRUPACION 3  
CLI.DOMICILIO,  
  
-- CODIGO POSTAL SEGUN AGRUPACION 3  
CLI.CP CODIGO_POSTAL,  
  
CLI.AGRU_3 AS AGRUPACION,  
AGR_CLI.DESCRIP_AGRU AS NOMBRE_AGRUPACION,  
CONCAT (ITM.TIPO_FACT,'-',RIGHT (CONCAT ('0000000000', ITM.NUM_FACT),8)) AS FACTURA_NRO,  
CONVERT (DATE, COM.FECHA,13) as FECHA,  
MONTH (COM.FECHA) as MES,  
  
-- TIPO DE ARTICULO SEGUN LETRA DE COMIENZO  
-- TIPO DE ARTICULO SEGUN LETRA DE COMIENZO  
ISNULL (art.COD_PROVEEDOR , ITM.ARTICULO ) AS COD_ARTICULO,  
ITM.ARTICULO AS COD_ARTICULO_RP,  
  
ART.DESCRIP_ARTI AS DESCRIPCION,  
ART.COD_BARRAS AS EAN,  
'' TIPO_DE_ARTICULO,  
  
-- SI EL PRODUCTO TIENE COMPONENTES. NO TIENE CANTIDAD, PORQUE SE DESCUENTA DE LOS PRODUCTOS
```

```

CASE WHEN ART.USA_COMPO  = 'S' THEN 1
      ELSE ITM.CANT * -1 END AS CANTIDAD,
ITM.CANT * PRECIO_NETO * -1 AS IMPORTE_S_IVA,
SUBSTRING(ITM.TIPO_FACT,1,1) AS TIPO_FACTURA,

'' MARKETPLACE,

-- SE TIENE QUE FILTRAR POR PROVEEDOR, ARMAR EL CONTROL DE PRODUCTOS SIN PROVEEDOR

0 AS PROVEEDOR,
ART.USA_COMPO,
com.fecha fecha_datetime ,
TIPO_COMP.COD_COMP ,
COM.CODI_VENDE ,
TIPO_COMP.COD_GRUPO ,
ART.AGRU_1 ,
ART.AGRU_2 ,
ART.SIN_STOCK ,
ART.UM_CONVER_COMPRA,
ITM.RENGLON ,
cli.CODI_ANTE ,
cli.AGRU_1 cli_agru_1,
itm.TIPO_FACT ,
itm.NUM_FACT ,
TIPO_COMP.PREFIJO

FROM dbo.RENG_FAC AS ITM LEFT JOIN dbo.COMP_EMITIDOS AS COM
ON ITM.NUM_FACT = COM.NUM AND ITM.TIPO_FACT = COM.TIPO

LEFT JOIN dbo.ARTICULOS AS ART
ON ITM.ARTICULO = ART.COD_ARTICULO

LEFT JOIN dbo.CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

LEFT JOIN dbo.COMP AS TIPO_COMP
ON ITM.TIPO_FACT = TIPO_COMP.COD_COMP

/*LEFT JOIN dbo.PROVE_ARTI AS PROV_ART
ON PROV_ART.ARTICULO = ITM.ARTICULO AND PROVE_PRINCIPAL = 'S'*/



LEFT JOIN dbo.AGRUP_CLIENTES AS AGR_CLI
ON CLI.AGRU_1 = AGR_CLI.CODI_AGRU AND AGR_CLI.NUM_AGRU = 1

WHERE COM.ESTADO != 'ANU'
AND LEFT(ITM.TIPO_FACT,1) in ('F','C')--, 'B')
and ITM.COMPO_ITEM is null
--and ( com.actua_stock = 'S' or ITM.val_tipo is not null)
--AND COM.FECHA BETWEEN '2021-01-01' AND '2021-02-28'

```

GO

- **POWERBI_COBRANZA_VISTA**

```
ALTER view [dbo].[powerbi_cobranzas_vista]
as
    select c.TIPO , c.NUM , cc.DESCRIP DESCRIP_COMP , cc.PREFIJO PUNTO_VENTA, cc.COD_GRUPO ,
           c.fecha FECHA_COMPRO, c.cliente, cl.razon ,
           p.IMPORTE , p.MONE , p.MONE_COTI ,p.VALO_TIPO , v.DESCRIP_VALOR ,
           p.VALO_BANCO , p.BANCO_DESTI , p.CUENTA_NUM , p.TARJE_CODI , p.CUOTAS , p.VALO_FECHA

    from COMP_EMITIDOS c inner join pagos p on (c.TIPO = p.TIPO and c.NUM = p.NUM )
        inner join clientes cl on (c.cliente = cl.num_cliente)
        inner join VALORES v on (p.VALO_TIPO = v.CODI )
        inner join comp cc on (c.TIPO = cc.cod_comp)
    where left(c.tipo,1) in ('I','A')
        and c.ESTADO != 'ANU'
        and c.FECHA >= '20200101'
```

GO

- POWERBI_COMPRO_PEND_VISTA

```
ALTER view [dbo].[powerbi_compro_pend_vista]
as
-- grant select on powerbi_compro_pend_vista to dba_power_bi

select c.TIPO , c.NUM , cc.DESCRIP descrip_comp, c.FECHA , cl.RAZON ,
       c.TOTAL , c.SALDO , c.MONE , cl.NOM_FANTASIA , c.ESTADO ,
       c.CODI_VENDE , v.NOMBRE nombre_vendedor ,
       ag.DESCRIP_AGRU descrip_agru_1 ,
       c.CLIENTE

from COMP_EMITIDOS c inner join comp cc on (c.tipo = cc.COD_COMP )
inner join CLIENTES cl on (c.CLIENTE = cl.NUM_CLIENTE )
left join VENDEDORES v on (c.CODI_VENDE = v.CODI_VENDE )
left join AGRUP_CLIENTES ag on (cl.AGRU_1 = ag.CODI_AGRU and ag.NUM_AGRU = 1 )
where left(c.tipo,1) not in ('S','I','V')
and c.ESTADO != 'ANU'
and c.SALDO <> 0
```

GO

- POWERBI_COMPRO_VENTAS_VISTA

```
ALTER view [dbo].[powerbi_compro_ventas_vista]
as
-- select * from powerbi_compro_ventas_vista

-- grant select on powerbi_compro_ventas_vista to dba_power_bi

SELECT
c.TIPO + '-' + RIGHT ( '00000000' + convert(varchar,c.NUM),8) AS compro_nro,
c.fecha,
c.CLIENTE,
cl.razon razon_social,
isnull(cc.cod_grupo,'') grupo_compro,
isnull(pr.NOMBRE,'Sin Provincia') provincia,
isnull(v.NOMBRE,'Sin Vendedor') vendedor,
isnull(ac.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_1,
c.cod_i_vende,
cl.nom_fantasia,
```

```

--co.DESCRIP_ORI origen_cliente,
--co2.DESCRIP_ORI origen_comprobante,
isnull(ac2.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_2,
isnull(ac3.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_3,
c.tipo cod_comprobante,
g.descrip grupo_compro_descrip ,
c.actua_stock,
--protra.RAZON transportista,
c.TOTAL
FROM comp_emitidos c inner join clientes cl on (c.cliente = cl.num_cliente)
    inner join comp cc on (c.tipo = cc.cod_comp)
    left join vendedores v on (c.codi_vende = v.codi_vende)
    left join provincias pr on (c.provincia = pr.codi)
    left join agrup_clientes ac on (cl.agru_1 = ac.codi_agru and ac.num_agru = 1)
    left join agrup_clientes ac2 on (cl.agru_2 = ac2.codi_agru and ac2.num_agru = 2)
    left join agrup_clientes ac3 on (cl.agru_3 = ac3.codi_agru and ac3.num_agru = 3)
    left join comp_grupos g on (cc.cod_grupo = g.cod_grupo)
    left join clientes_origen co on (cl.codi_ori = co.CODI_ORI)
    --left join clientes_origen co2 on (c.codi_ori = co.CODI_ORI)
    --left join PROVEEDORES protra on (c.CODI_TRANSPO = protra.NUM_PROV )
WHERE c.estado != 'ANU'
and left(c.tipo,1) in ('F','C','B')
and c.fecha between
    convert(datetime,convert(varchar(6),dateadd(mm,-36,getdate()),112)+'01') and getdate()
--and c.fecha between dateadd(mm,-10,getdate()) and getdate()

```

GO

- POWERBI_HOJARUTA_VISTA

```

ALTER view [dbo].[powerbi_hojaruta_vista]
as

```

```

select c.NUM HOJA_NUME, c.FECHA FECHA_HR, c.CHOFER , ch.NOMBRE_CHOFER ,
ce.TIPO , ce.NUM , cc.DESCRIP DESCRIP_COMP , ce.CLIENTE , cl.RAZON , ce.TOTAL ,
ce.FECHA FECHA_COMPRO
from HOJA_RUTA_CABE c inner join HOJA_RUTA_ITEMS i on (c.NUM = i.NUM )
inner join COMP_EMITIDOS ce on (i.COMPRO_TIPO = ce.TIPO and i.COMPRO_NUM = ce.NUM )
inner join comp cc on (ce.TIPO = cc.cod_comp)
inner join CHOFERES ch on (c.CHOFER = ch.CODI_CHOFER )
inner join clientes cl on (ce.cliente = cl.num_cliente)
where c.FECHA >= '20200101'
GO

```

- POWERBI_LISTAS_VISTA

```

ALTER view [dbo].[powerbi_listas_vista]
as
-- select * from powerbi_listas_vista
select a.cod_articulo,
       a.descrip_arti,
       a.desc_adicional,
       a.um,
       a.FECHA_MODI_COSTO,
       a.precio_uni costo_uni,
       a1.DESCRIP_AGRU agru_1,
       a2.DESCRIP_AGRU agru_2,
       a3.DESCRIP_AGRU agru_3,
       lc.LISTA_DESCRIP ,
       l1.LISTA_CODI ,
       l1.precio_vta ,
       l1.porcen_ganan ,
       l1.FECHA_MODI fecha_modi_lista,

       a.activo,
       a.sin_stock

from articulos a
left join agrupaciones a1 on (a.agru_1 = a1.cod_agrupacion and a1.num_agrupacion = 1)
left join agrupaciones a2 on (a.agru_2 = a2.cod_agrupacion and a2.num_agrupacion = 2)
left join agrupaciones a3 on (a.agru_3 = a3.cod_agrupacion and a3.num_agrupacion = 3)
left join listas_items l1 on (a.cod_articulo = l1.articulo )
left join LISTAS_CABE lc on (l1.LISTA_CODI = lc.LISTA_CODI )

```

GO

- POWERBI_MOV_STOCK_VISTA

```
ALTER view [dbo].[powerbi_mov_stock_vista]
as

-- select * from powerbi_mov_stock_vista

-- grant select on powerbi_mov_stock_vista to dba_power_bi

SELECT
r.TIPO_FACT + '-' + RIGHT ('00000000' + convert(varchar,r.NUM_FACT),8) AS compro_nro,
c.fecha,
year(c.fecha) anio,
c.CLIENTE codigo_cli_prove,
cl.razon razon_social,
r.articulo,
a.descrip_arti,
a.DESC_ADICIONAL ,
isnull(r.UM_MOVI,a.um) um,
a.KG_POR_UNIDAD ,
r.CANT AS cantidad,
isnull(cc.cod_grupo,'') grupo_compro,
isnull(pr.NOMBRE,'Sin Provincia') provincia,
isnull(v.NOMBRE,'Sin Vendedor') vendedor,
isnull(aa.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_1,
isnull(aa2.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_2,
r.depo deposito,
isnull(r.codi_cenco,'Sin CENCO') centro_costo ,
r.costo_uni costo_uni,
(r.CANT )*r.costo_uni costo_total,
r.costo_uni * 1.21 costo_uni_con_iva,
(r.CANT )*r.costo_uni * 1.21 costo_total_con_iva,
c.codi_vende,
cl.nom_fantasia,
c.tipo cod_comprobante,
g.descrip grupo_compro_descrip ,
r.PORCEN_DESCUEN porcen_descuen_item,
c.PORCEN_DESCUEN porcen_descuen_cabecera,
r.PORCEN_IVA porcen_iva,
cc.DESCRIP compro_descrip,
isnull(r.val_num,0) remito_nro ,
protra.RAZON transportista
FROM comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num = r.num_fact)
```

```

inner join articulos a on (r.articulo = a.cod_articulo)
inner join clientes cl on (c.cliente = cl.num_cliente)
inner join comp cc on (c.tipo = cc.cod_comp)
left join vendedores v on (c.codi_vende = v.codi_vende)
left join provincias pr on (c.provincia = pr.codi)
left join agrupaciones aa on (a.agru_1 = aa.codi_agru and aa.num_agru = 1)
left join agrupaciones aa2 on (a.agru_2 = aa2.codi_agru and aa2.num_agru = 2)
left join comp_grupos g on (cc.cod_grupo = g.cod_grupo)
left join PROVEEDORES protra on (c.CODI_TRANSPO = protra.NUM_PROV )

WHERE c.estado != 'ANU'
and left(c.tipo,1) in ('F','C','B','V')
and c.ACTUA_STOCK = 'S'
and c.fecha between convert(datetime,convert(varchar(6),dateadd(mm,-24,getdate())),112)+'01' ) and getdate()
--and c.fecha between dateadd(mm,-10,getdate()) and getdate()

```

UNION

```

SELECT
r.TIPO_FACT + '-' + RIGHT ( '00000000' + convert(varchar,r.NUM_FACT),8) AS compro_nro,
c.fecha,
year(c.fecha) anio,
c.CLIENTE codigo_cli_prove,
cl.razon_razon_social,
r.articulo,
a.descrip_arti,
a.DESC_ADICIONAL ,
isnull(r.UM_MOVI,a.um) um,
a.KG_POR_UNIDAD ,
r.CANT AS cantidad,
isnull(cc.cod_grupo,'') grupo_compro,
isnull(pr.NOMBRE,'Sin Provincia') provincia,
'Sin Vendedor' vendedor,
isnull(aa.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_1,
isnull(aa2.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_2,
r.depo deposito,
isnull(r.codi_cenco,'Sin CENCO') centro_costo ,
r.costo_uni costo_uni,
(r.CANT )*r.costo_uni costo_total,
r.costo_uni * 1.21 costo_uni_con_iva,
(r.CANT )*r.costo_uni * 1.21 costo_total_con_iva,
c.codi_vende,
cl.nom_fantasia,
c.tipo cod_comprobante,

```

```

g.descrip grupo_compro_descrip ,
r.PORCEN_DESCUEN porcen_descuen_item,
c.PORCEN_DESCUEN porcen_descuen_cabecera,
r.PORCEN_IVA porcen_iva,
cc.DESCRIP compro_descrip,
isnull(r.val_num,0) remito_nro ,
NULL transportista
FROM comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num = r.num_fact)
    inner join articulos a on (r.articulo = a.cod_articulo)
    inner join comp cc on (c.tipo = cc.cod_comp)
    left join PROVEEDORES cl on (c.cliente = cl.NUM_PROV)
    left join provincias pr on (c.provincia = pr.codi)
    left join agrupaciones aa on (a.agru_1 = aa.codi_agru and aa.num_agru = 1)
    left join agrupaciones aa2 on (a.agru_2 = aa2.codi_agru and aa2.num_agru = 2)
    left join comp_grupos g on (cc.cod_grupo = g.cod_grupo)
WHERE c.estado != 'ANU'
    and left(c.tipo,1) = 'S'
    and c.ACTUA_STOCK = 'S'
    and c.fecha between convert(datetime,convert(varchar(6),dateadd(mm,-24,getdate())),112)+'01' ) and
getdate()

```

GO

POWERBI_PAGOS_VISTA

```

ALTER view [dbo].[powerbi_pagos_vista]
as
select c.TIPO , c.NUM , cc.DESCRIP DESCRIP_COMP , cc.PREFIJO PUNTO_VENTA, cc.COD_GRUPO ,
c.fecha FECHA_COMPRO, c.PROVE , cl.razon ,
p.IMPORTE , p.MONE , p.MONE_COTI ,p.VALO_TIPO , v.DESCRIP_VALOR ,

```

```

p.VAL0_BANCO , p.CUENTA_NUM , p.VAL0_FECHA

from COMPRO_CABE_PROVE c inner join PAGOS_PROVE p on (c.TIPO = p.TIPO and c.NUM = p.NUM )
inner join PROVEEDORES cl on (c.PROVE = cl.NUM_PROV )
inner join VALORES v on (p.VAL0_TIPO = v.CODI )
inner join comp cc on (c.TIPO = cc.cod_comp)
where left(c.tipo,1) in ('H','N')
and c.ESTADO != 'ANU'
and c.FECHA >= '20200101'

```

GO

POWERBI_STOCK_VISTA

```

ALTER view [dbo].[powerbi_stock_vista]
as
-- grant select on powerbi_stock_vista to dba_power_bi
-- select * from powerbi_stock_vista

select a.COD_ARTICULO , a.DESCRIP_ARTI , a.DESC_ADICIONAL , a.UM ,
       a.FECHA_MODI_COSTO , a.PRECIO_UNI costo_uni ,
       ag.DESCRIP_AGRU descrip_agru_1,
       ag2.DESCRIP_AGRU descrip_agru_2,
       isnull(d.cant_stock,0) cant_stock,
       d.DEPO deposito,
       a.CODIGO_PRD ,
       a.DIAS_AVISO ,
       dd.DESCRIP descrip_deposito
from ARTICULOS a left join articulos_stock_depo d on (a.COD_ARTICULO = d.COD_ARTICULO )
left join DEPOSITOS dd on (d.DEPO = dd.DEPO_CODI )
left join AGRUPACIONES ag on (a.AGRU_1 = ag.CODI_AGRU and ag.NUM_AGRU = 1 )
left join AGRUPACIONES ag2 on (a.AGRU_2 = ag2.CODI_AGRU and ag2.NUM_AGRU = 2 )
where a.ACTIVO = 'S'
and a.SIN_STOCK = 'N'
/*group by a.COD_ARTICULO , a.DESCRIP_ARTI , a.DESC_ADICIONAL , a.UM ,
       a.FECHA_MODI_COSTO , a.PRECIO_UNI ,
       ag.DESCRIP_AGRU, ag2.DESCRIP_AGRU*/

```

GO

• POWERBI_VENTAS_VISTA

```
ALTER view [dbo].[powerbi_ventas_vista]
as

-- select * from powerbi_ventas_vista

-- grant select on powerbi_ventas_vista to cacho_power_bi

SELECT
r.TIPO_FACT + '-' + RIGHT ('00000000' + convert(varchar,r.NUM_FACT),8) AS compro_nro,
c.fecha,
year(c.fecha) anio,
c.CLIENTE,
cl.razon_razon_social,
r.articulo,
a.descrip_arti,
r.CANT * -1 AS cantidad,
PRECIO_ORIGINAL_REMITO as Precio_original,
(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) precio_neto,
(r.CANT * -1)*(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) sub_total,
(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) * 1.21 precio_neto_con_iva,
(r.CANT * -1)*(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) * 1.21 sub_total_con_iva,
isnull(cc.cod_grupo,'') grupo_compro,
isnull(pr.NOMBRE,'Sin Provincia') provincia,
isnull(v.NOMBRE,'Sin Vendedor') vendedor,
isnull(aa.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_1,
isnull(aa2.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_2,
isnull(aa3.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_3,
isnull(ac.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_1,
r.depo deposito,
isnull(r.codi_cenco,'Sin CENCO') centro_costo ,
--(r.CANT * -1)* ((r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) - r.costo_uni)
importe_margen,
r.costo_uni costo_uni,
(r.CANT * -1)*r.costo_uni costo_total,
r.costo_uni * 1.21 costo_uni_con_iva,
(r.CANT * -1)*r.costo_uni * 1.21 costo_total_con_iva,
c.codi_vende,
cl.nom_fantasia,
co.DESCRIP_ORI origen_cliente,
co2.DESCRIP_ORI origen_comprobante,
isnull(ac2.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_2,
```

```

isnull(ac2.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_3,
c.tipo cod_comprobante,
g.descrip grupo_compro_descrip ,
r.PORCEN_DESCUEN porcen_descuen_item,
c.PORCEN_DESCUEN porcen_descuen_cabecera,
r.PORCEN_IVA porcen_iva,
c.actua_stock,
isnull(r.val_num,0) remito_nro ,
protra.RAZON transportista ,
r.impor_impuess_total_vta as II_TOTAL,
a.porcen_impuess porcen_impuess_interno ,
r.PORCEN_IMPUES_INTER_HISTO ,
ori.DESCRIP_ORI descrip_origen,
0 PRECIO_LISTA_ORIGINAL,
r.PRECIO_NETO as precio_neto_ori -- CARGADO A PEDIDO DE ADRIAN PARA CONTROL

```

```

FROM comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num = r.num_fact)
    inner join articulos a on (r.articulo = a.cod_articulo)
    inner join clientes cl on (c.cliente = cl.num_cliente)
    inner join comp cc on (c.tipo = cc.cod_comp)
    left join vendedores v on (c.codi_vende = v.codi_vende)
    left join provincias pr on (c.provincia = pr.codi)
    left join agrupaciones aa on (a.agru_1 = aa.codi_agru and aa.num_agru = 1)
    left join agrupaciones aa2 on (a.agru_2 = aa2.codi_agru and aa2.num_agru = 2)
    left join agrupaciones aa3 on (a.agru_3 = aa3.codi_agru and aa3.num_agru = 3)
    left join agrup_clientes ac on (cl.agru_1 = ac.codi_agru and ac.num_agru = 1)
    left join agrup_clientes ac2 on (cl.agru_2 = ac2.codi_agru and ac2.num_agru = 2)
    left join agrup_clientes ac3 on (cl.agru_3 = ac3.codi_agru and ac3.num_agru = 3)
    left join comp_grupos g on (cc.cod_grupo = g.cod_grupo)
    left join clientes_origen co on (cl.codi_ori = co.CODI_ORI)
    left join clientes_origen co2 on (c.codi_ori = co2.CODI_ORI)
    left join PROVEEDORES protra on (c.CODI_TRANSPO = protra.NUM_PROV )
    left join CLIENTES_ORIGEN ori on (c.CODI_ORI = ori.CODI_ORI )
WHERE c.estado != 'ANU'
    and left(c.tipo,1) not in ('A','I','V','S')
    and c.fecha between convert(datetime,convert(varchar(6),dateadd(mm,-24,getdate())),112)+'01' ) and getdate()
    --and c.fecha between dateadd(mm,-10,getdate()) and getdate();

```

GO

Tablero Ventas

Cambios

- Poner en todos los tableros el descuento normalizado
-

PBI - DBA Ventas Capital

Definicion de margen

	Revenue		Costo		Cmg	Cmg%	Mkup %
	Unitario	Total					
Venta Sin Dto	8,175.00	98,100.00					
Dto 10%	7,357.50	88,290.00	5,880.98	70,571.78	17,718.22	20%	25%
II	1,323.54	15,882.49	1,323.54	15,882.49			
Base IVA	6,033.96	72,407.51	4,557.44	54,689.29	17,718.22	24%	32%
Base sin Impuestos	4,986.74	59,840.91	3,766.48	45,197.76	14,643.15	24%	32%

Caso FC F8Y-00001410

[DBA CENTRO CASO DE MARGEN \(003\).xlsx](#)

Este caso es el que usamos para reconstruir el costo desde la base y el % de II para mostrar los márgenes y Mkup que se obtiene con la venta

Tablero DBA / DBA Norte - Fuentes

Tablas

V-powerbi_ventas_vista

Reporta las ventas

```
SELECT
    r.TIPO_FACT + '-' + RIGHT ('00000000' + convert(varchar,r.NUM_FACT),8) AS compro_nro,
    c.fecha,
    year(c.fecha) anio,
    c.CLIENTE,
    cl.razon_razon_social,
    r.articulo,
    a.descrip_arti,
    r.CANT * -1 AS cantidad,
    PRECIO_ORIGINAL_REMITO as Precio_original,
    (r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) precio_neto,
    (r.CANT * -1)*(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) sub_total,
    (r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) * 1.21 precio_neto_con_iva,
    (r.CANT * -1)*(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) * 1.21 sub_total_con_iva,
    isnull(cc.cod_grupo,'') grupo_compro,
    isnull(pr.NOMBRE,'Sin Provincia') provincia,
    isnull(v.NOMBRE,'Sin Vendedor') vendedor,
    isnull(aa.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_1,
    isnull(aa2.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_2,
    isnull(aa3.DESCRIP_AGRU,'Sin Agrupacion') agrupacion_3,
    isnull(ac.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_1,
    r.depo deposito,
    isnull(r.codi_cenco,'Sin CENCO') centro_costo ,
    --(r.CANT * -1)* ((r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100)) - r.costo_uni) importe_margen,
    r.costo_uni costo_uni,
    (r.CANT * -1)*r.costo_uni costo_total,
    r.costo_uni * 1.21 costo_uni_con_iva,
    (r.CANT * -1)*r.costo_uni * 1.21 costo_total_con_iva,
    c.codi_vende,
    cl.nom_fantasia,
    co.DESCRIP_ORI origen_cliente,
    co2.DESCRIP_ORI origen_comprobante,
    isnull(ac2.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_2,
    isnull(ac2.DESCRIP_AGRU,'Sin Agrupacion') agrup_cliente_3,
    c.tipo cod_comprobante,
    g.descrip grupo_compro_descrip ,
    r.PORCEN_DESCUEN porcen_descuen_item,
    c.PORCEN_DESCUEN porcen_descuen_cabecera,
    r.PORCEN_IVA porcen_iva,
    c.actua_stock,
```

```

isnull(r.val_num,0) remito_nro ,
protra.RAZON transportista ,
r.importe_impuess_total_vta * dbl.SIGNO as II_TOTAL,
a.porcen_impuess porcen_impuess_interno ,
r.PORCEN_IMPUES_INTER_HISTO ,
ori.DESCRIP_ORI descrip_origen,
0 PRECIO_LISTA_ORIGINAL,
r.PRECIO_NETO as precio_neto_ori, -- CARGADO A PEDIDO DE ADRIAN PARA CONTROL
(((r.CANT * -1)*(r.PRECIO_NETO - (r.PRECIO_NETO*c.porcen_descuen/100))) * r.PORCEN_IVA /100) as iva
FROM comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num = r.num_fact)
    inner join articulos a on (r.articulo = a.cod_articulo)
    inner join clientes cl on (c.cliente = cl.num_cliente)
    inner join comp cc on (c.tipo = cc.cod_comp)
    left join vendedores v on (c.codi_vende = v.codi_vende)
    left join provincias pr on (c.provincia = pr.codi)
    left join agrupaciones aa on (a.agru_1 = aa.codi_agru and aa.num_agru = 1)
    left join agrupaciones aa2 on (a.agru_2 = aa2.codi_agru and aa2.num_agru = 2)
    left join agrupaciones aa3 on (a.agru_3 = aa3.codi_agru and aa3.num_agru = 3)
    left join agrup_clientes ac on (cl.agru_1 = ac.codi_agru and ac.num_agru = 1)
    left join agrup_clientes ac2 on (cl.agru_2 = ac2.codi_agru and ac2.num_agru = 2)
    left join agrup_clientes ac3 on (cl.agru_3 = ac3.codi_agru and ac3.num_agru = 3)
    left join comp_grupos g on (cc.cod_grupo = g.cod_grupo)
    left join clientes_origen co on (cl.codi_ori = co.CODI_ORI)
    left join clientes_origen co2 on (c.codi_ori = co2.CODI_ORI)
    Left join PROVEEDORES protra on (c.CODI_TRANSPO = protra.NUM_PROV )
    Left join CLIENTES_ORIGEN ori on (c.CODI_ORI = ori.CODI_ORI )
    LEFT JOIN DBA_VENDE_PBI DBI ON C.CODI_VENDE=DBI.CODI_VENDE
    LEFT JOIN DBA_COMP_PNL DBL ON C.TIPO=DBL.COD_COMP
WHERE c.estado != 'ANU'
    and left(c.tipo,1) not in ('A','I','V','S')
    and c.fecha between convert(datetime,convert(varchar(6),dateadd(mm,-24,getdate()),112)+'01') and
getdate()
    AND DBL.EN_PNL = 'S'
    AND DBI CONSIDERAR = 'S'
--and c.fecha between dateadd(mm,-10,getdate()) and getdate();

```

v-DBA_VENDE_PBI

Se usa para separar a los vendedores para considerar en el PBI, Cuales son plataformas y el tipo de canal

```
SELECT CODI_VENDE ,NOMBRE ,INACTIVO ,
CASE WHEN CODI_VENDE IN ('7','10','16','23','27') THEN 'N' ELSE 'S' END AS CONSIDERAR,
CASE WHEN CODI_VENDE IN ('9','37','40','51','60','65','61','62','63','52') THEN 'PLATAFORMAS'
ELSE 'ON' END AS TIPO_CANAL,
CASE WHEN CODI_VENDE = '9' THEN 'Ecommerce'
WHEN CODI_VENDE = '37' THEN 'Wabi'
WHEN CODI_VENDE = '40' THEN 'Tecnogo'
WHEN CODI_VENDE = '51' THEN 'Tecnogo ML'
WHEN CODI_VENDE = '60' THEN 'Tecnogo'
WHEN CODI_VENDE = '65' THEN 'Tecnogo Shopee'
WHEN CODI_VENDE = '61' THEN 'Tecnogo'
WHEN CODI_VENDE = '62' THEN 'Tecnogo Shopify'
WHEN CODI_VENDE = '63' THEN 'ICBC Mall'
WHEN CODI_VENDE = '52' THEN 'Tecnogo ML'
END AS TIPO_PLATAFORMA,
'dba' AS BASE
FROM VENDEDORES v
```

T-DBA_COMP_PNL

En esta tabla se ponen los comprobantes que se deben considerar dentro del reporte de ventas del PBI,

```
select *
from
DBA_COMP_PNL
where COD_COMP in ('FVA','FVB','NVA','NVB')
```

sql de insert para sumar los comprobantes que deben ser tenidos en cuenta en la extraccion

```
INSERT INTO DBA_COMP_PNL (COD_COMP,EN_PNL,SIGNO) VALUES ('FVA','S',1);
INSERT INTO DBA_COMP_PNL (COD_COMP,EN_PNL,SIGNO) VALUES ('FVB','S',1);
INSERT INTO DBA_COMP_PNL (COD_COMP,EN_PNL,SIGNO) VALUES ('NVA','S',-1);
INSERT INTO DBA_COMP_PNL (COD_COMP,EN_PNL,SIGNO) VALUES ('NVB','S',-1);
```

Cashflow Administracion

Diagrama explicativo

Diagrama

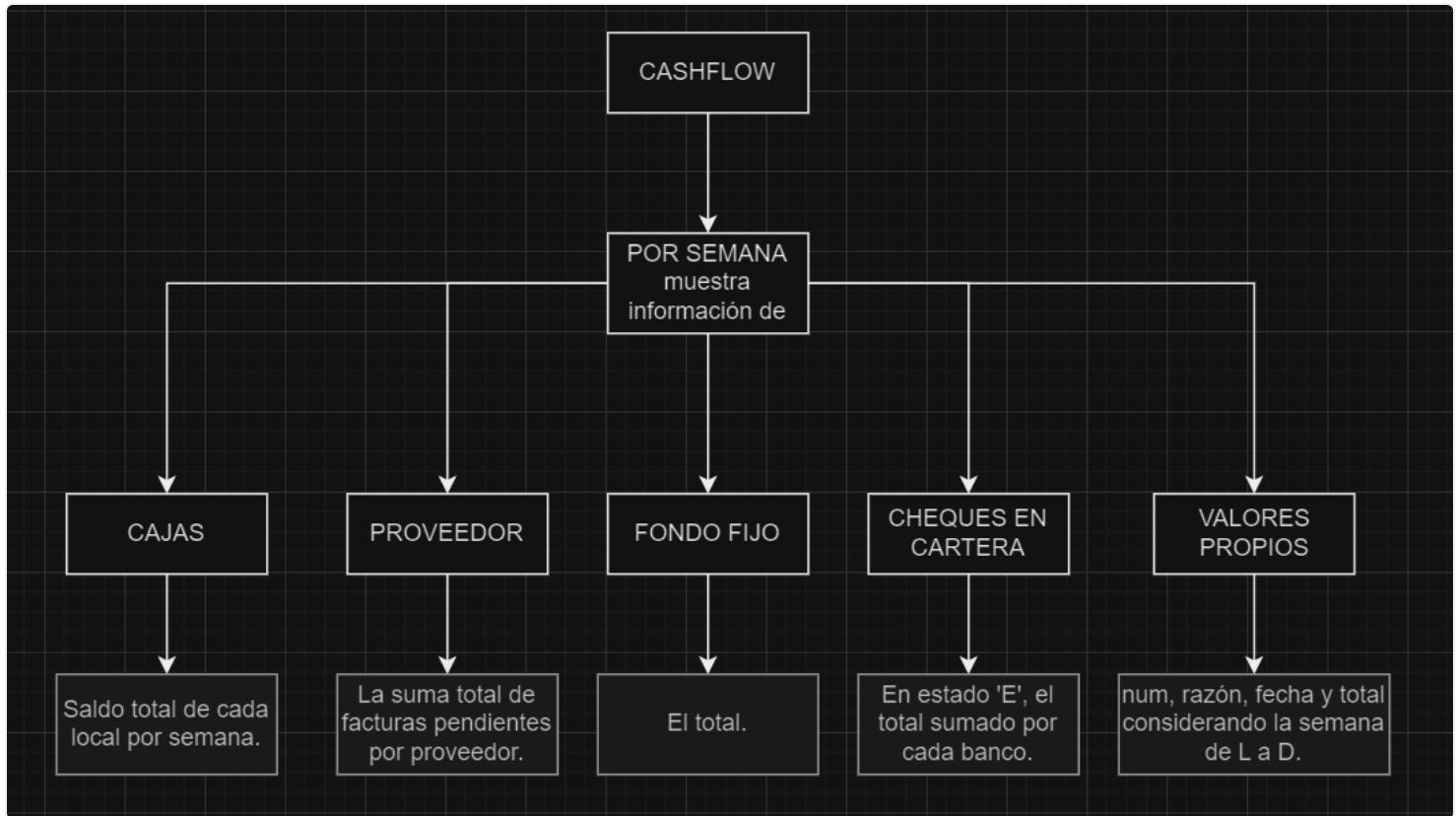
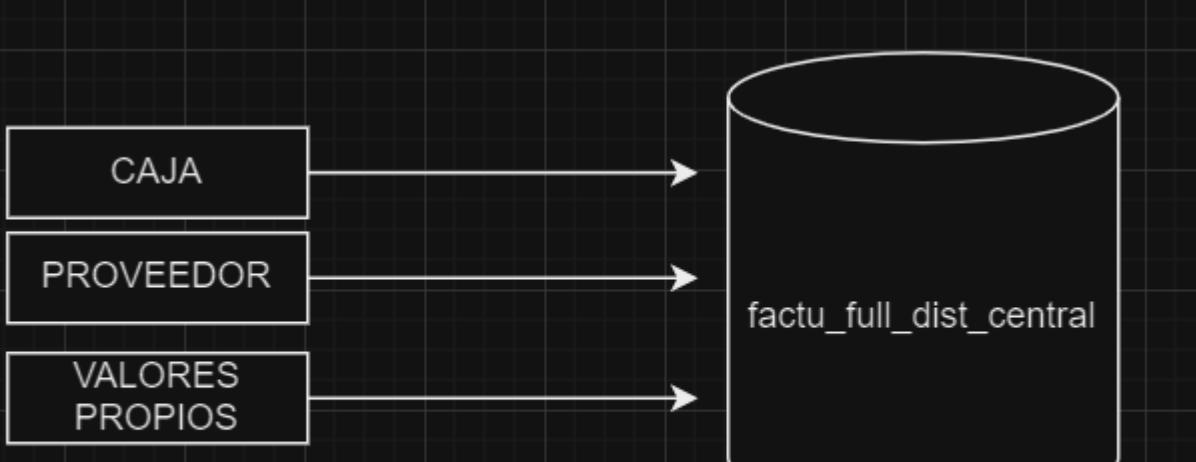
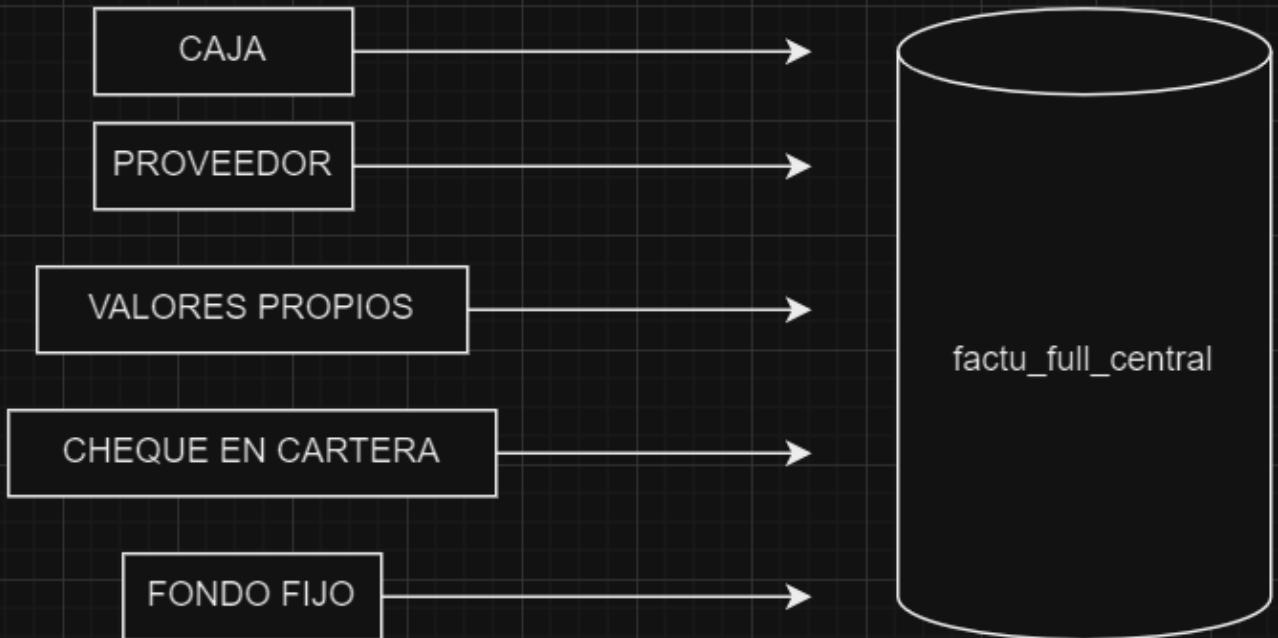


Diagrama extracción de datos por base:



SQL Saldo caja

```
select c.fecha, c.USUA_APER , i.CODI_CONCEP , cs.DESCRIP_CONCEP , i.IMPOR * cs.SIGNO as PESOS, 0 as USD,
i.MONE_COTI
from caja_cabe as c left join caja_items as i
    on c.FECHA = i.FECHA and c.VERSION = i.VERSION
    left join CONCEPTOS as cs
        on i.CODI_CONCEP = cs.CODI_CONCEP
where c.fecha = '2023-08-22'
and i.MONE = 'PES'
and c.USUA_APER = 'fel_palermo'

union all

select c.fecha, c.USUA_APER , i.CODI_CONCEP , cs.DESCRIP_CONCEP , i.IMPOR * i.MONE_COTI as PESOS,
i.IMPOR * cs.SIGNO as USD, i.MONE_COTI
from caja_cabe as c left join caja_items as i
    on c.FECHA = i.FECHA and c.VERSION = i.VERSION
    left join CONCEPTOS as cs
        on i.CODI_CONCEP = cs.CODI_CONCEP
where c.fecha = '2023-08-22'
and i.MONE = 'U$S'
and c.USUA_APER = 'fel_palermo'
```

@Ailen te dejo aca el sql

Store procedure creado que muestra el saldo final actual de las cajas de central y distribuidora central.

```
use factu_full_central
EXEC cashflow_caja_consulta
```

```
ALTER PROCEDURE dbo.cashflow_caja_consulta
(
    @fecha_desde datetime = NULL,
    @fecha_hasta datetime = NULL
)
AS
```

```
CREATE TABLE #general
(
    codi_banco varchar (20),
    cuenta_num varchar (20),
    mone varchar (20),
    impor varchar (20),
    valo_fecha datetime,
    valo_num varchar (20),
    valo_tipo varchar (20),
    valo_descrip varchar (20),
    valo_banco_ori varchar (20),
    compro_tipo varchar (20),
    compro_num varchar (20),
    compro_modu varchar (20),
    compro_descrip varchar (20),
    clien_prove varchar (20),
    fecha_real varchar (20),
    obser varchar (20),
    saldo_ini varchar (20),
    descrip_banco varchar (20),
    fecha_alta varchar (20)
)
```

```
CREATE TABLE #general2
(
    codi_banco varchar (20),
    cuenta_num varchar (20),
    mone varchar (20),
    impor varchar (20),
    valo_fecha datetime,
    valo_num varchar (20),
    valo_tipo varchar (20),
    valo_descrip varchar (20),
    valo_banco_ori varchar (20),
    compro_tipo varchar (20),
    compro_num varchar (20),
    compro_modu varchar (20),
    compro_descrip varchar (20),
    clien_prove varchar (20),
    fecha_real varchar (20),
    obser varchar (20),
    saldo_ini varchar (20),
    descrip_banco varchar (20),
    fecha_alta varchar (20)
)
```

```
DECLARE @fecha_actual DATETIME
SET @fecha_actual = GETDATE()
```

```

insert into #general (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num,
valo_tipo, valo_descrip, valo_banco_ori, compro_tipo, compro_num,
compro_modu, compro_descrip, clien_prove, fecha_real, obser, saldo_ini,
descrip_banco, fecha_alta)

EXEC factu_full_central.dbo.factu_saldos_banca_lista @fecha_desde = @fecha_actual, @fecha_hasta =
@fecha_actual, @codi_banco = 'CAJ', --@CUENTA_NUM = 'CAJA FLORESTA',
@FECHA_TIPO = 'S', @agregar_mov_auxi = 'N', @cuenta_tipo = 'T', @agrupar_compro = 'N',
@es_admin = 'S', @usuario = 'full_sa', @cod_grupo = 'NULL'

insert into #general2 (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num,
valo_tipo, valo_descrip, valo_banco_ori, compro_tipo, compro_num,
compro_modu, compro_descrip, clien_prove, fecha_real, obser, saldo_ini,
descrip_banco, fecha_alta)

EXEC factu_full_dist_central.dbo.factu_saldos_banca_lista @fecha_desde = @fecha_actual, @fecha_hasta =
@fecha_actual, @codi_banco = 'CAJ', --@CUENTA_NUM = 'CAJA FLORESTA',
@FECHA_TIPO = 'S', @agregar_mov_auxi = 'N', @cuenta_tipo = 'T', @agrupar_compro = 'N',
@es_admin = 'S', @usuario = 'full_sa'--, @cod_grupo = 'NULL'

SELECT
  'FK ' + g.cuenta_num CAJA,
  CAST(saldo_ini AS decimal(18,2)) + CAST(impor AS decimal(18,2)) AS SALDO_FF
FROM #general g
WHERE g.cuenta_num like 'CAJA%'

union
SELECT
  'DC ' + g2.cuenta_num CAJA,
  CAST(saldo_ini AS decimal(18,2)) + CAST(impor AS decimal(18,2)) AS SALDO_FF
FROM #general2 g2
WHERE g2.cuenta_num like 'CAJA%'

DROP TABLE #general
DROP TABLE #general2

```

GO

Nota

Debe y haber

```
--DEBE Y HABER DE CAJAS. Perfecto
exec factu_saldosConta_lista @fecha_desde = '2024-02-28', @fecha_hasta = '2024-02-28'
-- 11111, 11111M0, 11112, 11118
```

Creado

```
use factu_full_central
exec cashflow_factu_saldos_banca_lista @fecha_desde = '2024-02-28', @fecha_hasta = '2024-02-28'
```

Original

```
use factu_full_central
exec factu_saldos_banca_lista @fecha_desde = '2024-02-28', @fecha_hasta = '2024-02-28'
```

SQL Proveedor V.2

```
USE factu_full_central or USE factu_full_dist_central

EXEC cashflow_proveedor_consulta
```

Store Procedure creado en factu_full_dist_central y factu_full_central

cashflow_proveedor_consulta

```
ALTER PROCEDURE [dbo].[cashflow_proveedor_consulta](
@fecha_desde_pago date = NULL,
@fecha_hasta_pago date = NULL
--@agru_prove varchar (3)
)

/*
    NOTA: se eliminaron las conversiones porque cuando aplicamos formulas en excel no toman el valor
    de las celdas
*/
as
DECLARE @SemanaActual INT
SET @SemanaActual = DATEPART(WEEK, GETDATE()) --semana 9
SELECT
    PR.NUM_PROV AS NUM,
    PR.RAZON AS PROVEEDOR,
    C.TIPO AS COD,
    CP.DESCRIP AS COMPROBANTE,
    C.NUM AS NUM_COMP,
    C.SUCUR,
    C.ESTADO,
    C.FECHA_PAGO AS FECHA_PAGO,
    CASE --Evaluar número semana del mes
        --Si el año es menor al actual y la semana menor o si directamente el año es menor al
        actual, va 0
        WHEN ( (DATEPART(YEAR, C.FECHA_PAGO) < DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK,
C.FECHA_PAGO) < DATEPART(WEEK, GETDATE())) )
            OR (DATEPART(YEAR, C.FECHA_PAGO) < DATEPART(YEAR, GETDATE())) THEN 0
        --Si el año es igual al actual y la semana es menor a la actual, va también 0
        WHEN (DATEPART(YEAR, C.FECHA_PAGO) = DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK,
C.FECHA_PAGO) < DATEPART(WEEK, GETDATE() )) THEN 0
        --Si el año es igual y la semana es mayor a la actual o el mes no es el mismo que el
        actual va 6
        WHEN (DATEPART(YEAR, C.FECHA_PAGO) = DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK,
```

```

C.FECHA_PAGO) > DATEPART(WEEK, GETDATE()) ) OR (DATEPART(MONTH, C.FECHA_PAGO) != DATEPART(MONTH,
GETDATE()))
THEN 6
--Si no cumple todo estos casos, quiere decir que estoy en el mes actual y la semana actual,
entonces muestro el número de la semana del mes actual
--considerando solo este mes. Es decir, la semana 1 de este mes, la semana 2 de este mes,
etc.
ELSE DATEPART(WEEK, GETDATE()) - DATEPART(WEEK, DATEADD(MM, DATEDIFF(MM, 0, GETDATE()), 0))
+ 1
--La semana que se va a mostrar
END AS semana_mes,
C.TOTAL AS TOTAL_COMP,
SUM (C.TOTAL) OVER (PARTITION BY PR.NUM_PROV) AS SUMA_TOTAL

FROM COMPRO_CABE_PROVE AS C
INNER JOIN COMP AS CP ON (C.TIPO = CP.COD_COMP)
INNER JOIN PROVEEDORES AS PR ON (C.PROVE = PR.NUM_PROV)
WHERE C.SALDO > 0
AND C.ESTADO = 'PEN'
AND PR.ESTADO = 'A'
AND (COD_COMP = 'PFA' OR COD_COMP = 'PFN') --Factura A y Factura Interna
AND (C.TIPO = 'PFA' OR C.TIPO = 'PFN') --Factura A y Factura Interna
--AND C.FECHA_PAGO BETWEEN @fecha_desde_pago AND @fecha_hasta_pago
--AND C.VENCIMIENTO > GETDATE() --no sería necesario.
AND C.TOTAL != 0.01
--AND PR.AGRU_1 = @agru_prove
ORDER BY C.FECHA_PAGO

```

GO

Explicación

- Se limita a comprobantes con saldo mayor a 0

```
WHERE C.SALDO > 0
```

- Se limita a comprobantes con estado pendiente

```
AND C.ESTADO = 'PEN'
```

- Se considera proveedores activos

```
AND PR.ESTADO = 'A'
```

- Se considera factura interna y tipo a

```
AND (COD_COMP = 'PFA' OR COD_COMP = 'PFN')
```

- Se excluye los comprobantes que tienen total 0.01

```
AND C.TOTAL != 0.01
```

- Se ordena por fecha de pago

```
ORDER BY C.FECHA_PAGO
```

Explicación estructura CASE

- Si el año es menor al actual y la semana menor, o si directamente, el año es menor al actual, va 0

```
WHEN ( (DATEPART(YEAR, C.FECHA_PAGO) < DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK, C.FECHA_PAGO) < DATEPART(WEEK, GETDATE()))) )  
OR (DATEPART(YEAR, C.FECHA_PAGO) < DATEPART(YEAR, GETDATE())) THEN 0
```

- Si el año es igual al actual y la semana es menor a la actual, va también 0

```
WHEN (DATEPART(YEAR, C.FECHA_PAGO) = DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK, C.FECHA_PAGO) < DATEPART(WEEK, GETDATE()) ) THEN 0
```

- Si el año es igual y la semana es mayor a la actual o el mes no es el mismo que el actual va 6

```
WHEN (DATEPART(YEAR, C.FECHA_PAGO) = DATEPART(YEAR, GETDATE()) AND DATEPART(WEEK, C.FECHA_PAGO) > DATEPART(WEEK, GETDATE()) OR (DATEPART(MONTH, C.FECHA_PAGO) != DATEPART(MONTH, GETDATE()))) THEN 6
```

- Si no cumple todo estos casos, quiere decir que estoy en el mes actual y la semana actual, entonces muestro el número de la semana del mes actual, considerando solo este mes. Es decir, la semana 1 de este mes, la semana 2 de este mes, etc.

```
ELSE DATEPART(WEEK, GETDATE()) - DATEPART(WEEK, DATEADD(MM, DATEDIFF(MM, 0, GETDATE()), 0)) + 1
```

SQL Versión 1

Muestra el mes del comprobante y en base a eso, indica la semana del mes del comprobante

```
ALTER PROCEDURE [dbo].[cashflow_proveedor_consulta]
(
@fecha_desde_pago date = NULL,
@fecha_hasta_pago date = NULL
--@agru_prove varchar (3)
)
as
SELECT
PR.NUM_PROV AS NUM_PROVE,
PR.RAZON AS PROVEEDOR,
C.TIPO AS COD_COM,
CP.DESCRIP as COMPROBANTE,
C.NUM AS NUM_COMP,
C.SUCUR,
--(CONVERT (varchar, C.FECHA_PAGO,104)) AS FECHA_PAGO,
--se eliminaron las conversiones porque cuando aplicamos formulas en excel no toman el valor de
las celdas
C.FECHA_PAGO AS FECHA_PAGO,
CASE --Traducir el mes a español
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'January' THEN 'Enero'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'February' THEN 'Febrero'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'March' THEN 'Marzo'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'April' THEN 'Abril'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'May' THEN 'Mayo'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'June' THEN 'Junio'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'July' THEN 'Julio'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'August' THEN 'Agosto'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'September' THEN 'Septiembre'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'October' THEN 'Octubre'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'November' THEN 'Noviembre'
    WHEN DATENAME(MONTH, C.FECHA_PAGO) = 'December' THEN 'Diciembre'
END AS MES, --Qué mes
(DATEPART(WEEK, C.FECHA_PAGO) - DATEPART(WEEK, DATEADD(DAY, 1 - DATEPART(DAY, C.FECHA_PAGO),
C.FECHA_PAGO))) + 1 AS NUM_SEMANA_MES, --Qué semana del mes
C.ESTADO,
--'$ ' + CONVERT(VARCHAR, CONVERT(VARCHAR, CAST(C.TOTAL AS MONEY), 1)) AS TOTAL_COMP,
--'$ ' + CONVERT(VARCHAR, CONVERT(VARCHAR, CAST(SUM (C.TOTAL) OVER (PARTITION BY PR.NUM_PROV) AS
MONEY),1)) AS SUMA_TOTAL
C.TOTAL AS TOTAL_COMP,
SUM (C.TOTAL) OVER (PARTITION BY PR.NUM_PROV) AS SUMA_TOTAL
```

```
FROM COMPRO_CABE_PROVE AS C
    INNER JOIN COMP AS CP ON (C.TIPO = CP.COD_COMP)
    INNER JOIN PROVEEDORES AS PR ON (C.PROVE = PR.NUM_PROV)
WHERE C.SALDO > 0
    AND C.ESTADO = 'PEN'
    AND PR.ESTADO = 'A'
    AND (COD_COMP = 'PFA' OR COD_COMP = 'PFN') --Factura A y Factura Interna
    AND (C.TIPO = 'PFA' OR C.TIPO = 'PFN') --Factura A y Factura Interna
    --AND C.FECHA_PAGO BETWEEN @fecha_desde_pago AND @fecha_hasta_pago
    --AND C.VENCIMIENTO > GETDATE() --no sería necesario.
    AND C.TOTAL != 0.01
    --AND PR.AGRU_1 = @agru_prove
ORDER BY C.FECHA_PAGO
```

GO

En Excel

En Excel se pone la consulta en una Tabla Dinámica

Se puede diferenciar por agrupación del proveedor (SER o PRO)

0 ☐ siempre son facturas viejas a la semana actual.

6 ☐ siempre son facturas de semanas siguientes.

1, 2, 3, 4, 5 ☐ siempre corresponderán a la semana actual del mes según corresponda.

Proveedor - Central					
AGRUPACION_PROVE	PRO	0	5	6	Total general
Suma de TOTAL_COMP	Semana				
PROVEEDOR		0	5	6	Total general
+ BODEGA NORTON S.A.	\$ 3,774,735.98			\$ 3,774,735.98	
+ BODEGAS CHANDON S A	\$ 16,420,205.80		\$ 43,912,805.49	\$ 60,333,011.29	
+ BODEGAS SALENTEIN SA	\$ 495,886.43		\$ 881,652.28	\$ 1,377,538.71	
+ CAMPARI ARGENTINA S A	\$ 9,281,108.12	\$ 21,429,351.29	\$ 30,710,459.41		
+ CEPAS ARGENTINAS S. A.	\$ 8,149,719.16	\$ 545,858.76	\$ 6,627,784.15	\$ 15,323,362.07	
+ Consultora Umami SRL	\$ 1,629,551.90			\$ 1,629,551.90	
+ COSMOPOLITAN SOCIEDAD ANONIMA		\$ 2,069,516.11	\$ 2,069,516.11		
+ DELLEPIANE SAN LUIS S.A	\$ 12,159,756.69			\$ 12,159,756.69	
+ DESTILERIA DEL SUR S.R.L	\$ 430,946.40			\$ 430,946.40	
+ Distribuidora Celestino SRL	\$ 134,265.60			\$ 134,265.60	
+ Finca Algarve S.A.	\$ 2,146,685.97			\$ 2,146,685.97	
+ Fratelli Branca Destilerias S.A.	\$ 7,417,274.23	\$ 29,238,790.13	\$ 36,656,064.36		
+ Grupo Peñaflor S.A.	\$ 23,831,411.76	\$ 40,747,798.30	\$ 64,579,210.06		
+ Javier Fuente	\$ 96,800.00			\$ 96,800.00	
+ KILOMETRO 342 S.A	\$ 397,441.44			\$ 397,441.44	
+ La Agricola S.A.	\$ 15,839,045.81	\$ 9,196,854.53	\$ 25,035,900.34		
+ LA RURAL VIÑEDOS Y BODEGAS S.A LTDA.	\$ 11,615,260.10			\$ 11,615,260.10	
+ LEONCIO ARIZU S A	\$ 4,202,543.51	\$ 2,519,001.79	\$ 6,721,545.30		
+ MITRE FORTIN S.R.L.	\$ 4,763,485.32	\$ 4,525,361.49	\$ 9,288,846.81		
+ Phaedrus SRL	\$ 6,226,993.69			\$ 6,226,993.69	
+ R&HL S.A.	\$ 5,175,919.35			\$ 5,175,919.35	
+ Stapler S.R.L.	\$ 177,724.80			\$ 177,724.80	
+ VAZQUEZ OMAR JOAQUIN	\$ 494,478.00			\$ 494,478.00	
Total general	\$ 125,580,131.94	\$ 9,826,966.88	\$ 161,148,915.56	\$ 296,556,014.38	

Opción dos:

Que filtre por el mes y de ese mes te divide por todas las semanas del mes

Proveedor - Dist. Central						
MES	Enero					
Proveedor	Semana	1	2	3	4	5 Total general
Suma de SUMA_TOTAL		\$ 271,398.00				\$ 271,398.00
Proveedor						
+ 2F SERVICIOS SAS		\$ 226,900.00				\$ 226,900.00
+ ANASTASIA MARTA BADALOVICH DE ALTUZARRA			\$ 2,677,346.42	\$ 5,354,692.84		\$ 8,032,039.26
+ DANLOVIC S.R.L			\$ 37,902.31			\$ 37,902.31
+ FARMACIA GENERAL PACHECO SCS				\$ 9,615,238.19		\$ 9,615,238.19
+ FERCHETTO SAS		\$ 30,419,280.66	\$ 26,451,548.40	\$ 17,193,506.46	\$ 34,387,012.92	\$ 18,516,083.88
+ GASTOS GENERALES SUCURSALES					\$ 16,500.02	\$ 16,500.02
+ Hanono Martin Leonardo						
+ MERCADOLIBRE S.R.L.		\$ 1,066,559.99				\$ 1,066,559.99
+ Panizza y Faccaro S.R.L			\$ 28,315.70			\$ 28,315.70
+ Rucamet SRL		\$ 19,549.00				\$ 19,549.00
+ SALGADO LUIS MARIANO			\$ 284,350.00			\$ 284,350.00
+ Telecom Argentina S.A.			\$ 3,221,726.00			\$ 3,221,726.00
+ TRASLAPIEDRA SAS		\$ 5,603,058.41	\$ 5,603,058.41			\$ 11,206,116.82
+ Vinecol S. A.		\$ 1,819,997.60				\$ 1,819,997.60
Total general		\$ 33,552,287.25	\$ 32,326,004.81	\$ 26,330,956.57	\$ 37,118,761.67	\$ 33,486,014.91
						\$ 162,814,025.21

Extracción desde Factu

Consultas > Proveedores > Composición de Saldos.

Se indica el proveedor a consultar y se busca. Considera solo Facturas A y Facturas Int.

The screenshot shows the Factu software interface with the following details:

- Menu Bar:** Consultas, Ventanas, Salir.
- Toolbars:** F, V, C, F, C, R, A, N, C, P, R, C, H, R, S, T, F, C, A, N, C, O, P, O, C.
- Search Bar:** Texto a Buscar, with icons for search, filter, and mail.
- Left Sidebar:** Contabilidad, Fondos, Proveedores, Ventas, Stock, Auditoria, Acerca de ..., ODBC.
- Central Panel:** Autorizaciones, Comprobantes Ingresados, Composición de Saldos (selected), Compras, IVA Compras, Ordenes de Compra, Proveedores, Retenciones Realizadas, Impuestos, Saldos de Proveedores, Valores Entregados.
- Filter Panel:** Filtrar por Rango de Fechas (Emisión), Desde: 27/02/2024, Hasta: 27/02/2024.
- Data Grid:** A table showing transaction details. The columns are: Nro., Estado, Moneda, Total, Saldo, Saldo Acum., F. Venci., Cotización, and Días FF.
- Table Data:** The grid contains approximately 15 rows of transaction data, such as Factura Int., Anticipo a Prove. Int., and N.C. Int., with their respective dates, amounts, and status.

Modificación de factura

Prototipo, desarrollado desde factu_full_cetral_desa_nacho Esperando confirmación para hacerlo en factu_full_central

Store procedure creado para modificar la fecha de pago de una factura seleccionada:

Se indica num, tipo, fecha pago, sucursal y proveedor para identificar la factura.

Maneja validaciones en caso de que el num, tipo, sucursal o proveedor que indique no exista.

No maneja la validación en caso de que indique la misma fecha de pago dos veces porque Excel ejecuta dos veces el store procedure, salta como error cuando no debería.

Cuando pasa las validaciones y realiza la modificación de la factura, muestra: 'La factura número ',@NUM, ' del proveedor ',@PROVE, ' se ha modificado.'

```
use factu_full_central_desa_nacho
EXEC modi_fact_prove_cashflow @NUM = '36', @TIPO='PFN', @FECHA_PAGO = '2024-03-07', @PROVE = '4', @SUCUR
= '2'
```

```
USE [factu_full_central_desa_nacho]
GO

***** Object: StoredProcedure [dbo].[modi_fact_prove_cashflow]      Script Date: 04/12/2023 15:38:04
*****
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

/*
Store procedure creado para el CashFlow de área contabilidad (Belen)
Se indica num, tipo, fecha pago, sucursal y proveedor para identificar la factura
*/
ALTER PROCEDURE [dbo].[modi_fact_prove_cashflow] (
@NUM int = NULL,
```

```

@TIPO varchar(6) = NULL,
@FECHA_PAGO datetime = NULL,
@SUCUR int = NULL,
@PROVE int = NULL
)
AS
BEGIN --Inicio del cuerpo del store

    --EJEMPLO: UPDATE COMPRO_CABE_PROVE SET FECHA_PAGO = '2023-12-06' WHERE NUM = '36' AND TIPO = 'PFN'
    AND SUCUR = 2 AND PROVE = 4

    --Verifica que ninguna variable sea null

    IF @NUM IS NULL OR @TIPO IS NULL OR @FECHA_PAGO IS NULL OR @SUCUR IS NULL OR @PROVE IS NULL
        THROW 51000, 'Debe completar todos los campos para modificar la fecha de la factura', 1;

    --Verifica que el proveedor existe

    IF @PROVE IS NOT NULL AND NOT EXISTS (SELECT 1 FROM PROVEEDORES P WHERE P.NUM_PROV = @PROVE)--AND
    ESTADO = 'Activo'
        THROW 51000, 'El proveedor indicado, no existe', 2;

    --Verificar que fecha de la factura no es la misma a la ingresada

    --A veces se ejecuta 2 veces el store desde Excel y por eso aparece 2 veces.
    /*IF EXISTS ( -- si existe
        SELECT 1
        FROM COMPRO_CABE_PROVE
        WHERE NUM = @NUM
            AND TIPO = @TIPO
            AND SUCUR = @SUCUR
            AND PROVE = @PROVE
            AND CAST(FECHA_PAGO AS DATE) = CAST(@FECHA_PAGO AS DATE) -- Compara solo la parte
de fecha, ignorando la hora
    )
    THROW 51000, 'La fecha ingresada es la misma que la fecha actual de la factura.', 4;

    Excel ejecuta dos veces el store procedure entonces es complicado aplicar esta regla

*/
    --Verificar que el tipo de comprobante exista
    IF NOT EXISTS (SELECT 1 FROM COMPRO_CABE_PROVE WHERE TIPO = @TIPO)
        THROW 51000, 'El tipo de factura indicado no existe.', 5;

    --Verificar que el tipo de sucursal exista
    IF NOT EXISTS (SELECT 1 FROM COMPRO_CABE_PROVE WHERE SUCUR = @SUCUR)
        THROW 51000, 'La sucursal indicada no existe.', 5;

    --Verificar que el num exista exista

```

```
IF NOT EXISTS (SELECT 1 FROM COMPRO_CABE_PROVE WHERE NUM = @NUM)
    THROW 51000, 'El numero de factura indicado no existe.', 5;
```

--Si pasa las validaciones:

```
BEGIN --Inicio de modificar
```

```
UPDATE COMPRO_CABE_PROVE SET FECHA_PAGO = @FECHA_PAGO, --Modificar fecha de pago
    FECHA_MODI = GETDATE(),
    USER_MODI = 'excel_cashflow'
```

```
WHERE NUM = @NUM
```

```
AND TIPO = @TIPO
```

```
AND SUCUR = @SUCUR
```

```
AND PROVE = @PROVE
```

```
SELECT CONCAT('La factura número ',@NUM, ' del proveedor ',@PROVE,' se ha modificado.') as HECHO
```

```
END--Fin del modificar
```

```
END--Fin del cuerpo del Store
```

```
GO
```

En Excel: (Prototipo, desarrollado desde factu_full_cetral_desa_nacho Esperando confirmación para hacerlo en factu_full_central)

NUM	TIPO	FECHA_PAGO	SUCUR	PROVE	HECHO
36	PFN	2023-09-05	2	4	La factura numero 36 del proveedor 4 se ha modificado.

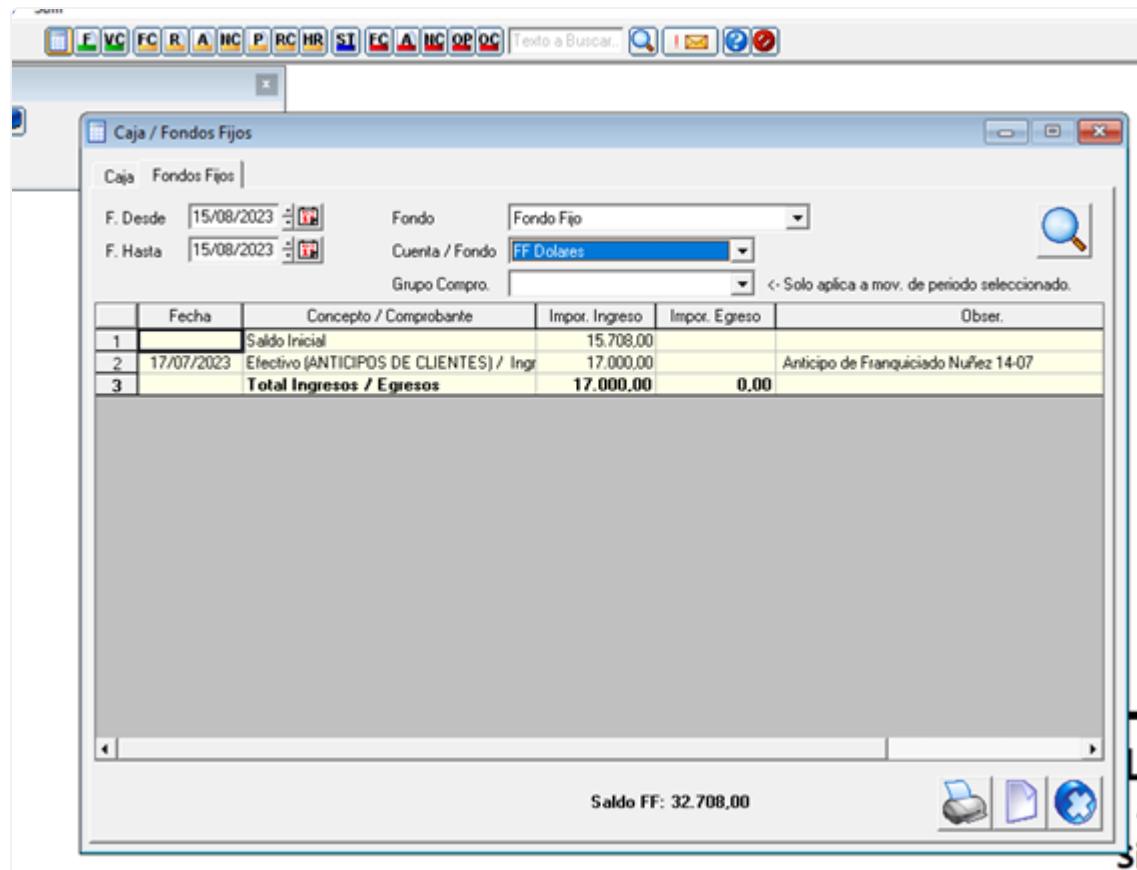
PAGO PROVEEDORES								Filtrar por fecha		
								Fecha_d	Fecha_hasta	COD_PROVE
								2023-09-10	2023-12-30	PRO
Pendientes										
NUM_PROVE	PROVEEDOR	COD_COM	COMPROBANTE	NUM_COMP	SUCUR	FECHA_PAGO	ESTADO	TOTAL_COMP	SUMA_TOTAL	
1	Cia Industrial Cerveceria S.A.	PFA	Factura A	33	1	02.11.2023	PEN	\$ 159,449.08	\$ 318,898.16	
1	Cia Industrial Cerveceria S.A.	PFN	Factura Int.	34	1	20.11.2023	PEN	\$ 159,449.08	\$ 318,898.16	
2	Pernod Ricard Argentina SRL	PFN	Factura Int.	36	1	09.10.2023	PEN	\$ 159,449.08	\$ 159,449.08	
4	Grupo Peñaflor S.A.	PFN	Factura Int.	36	2	05.09.2023	PEN	\$ 159,449.08	\$ 159,449.08	

SQL Fondo Fijo

Extracción desde el factu

1. Icono de la caja
2. Opción Fondo Fijos
3. Seleccionar en 'Fondo' Fondo Fijo
4. y en 'Cuenta / Fondo' FF Dolares

El campo que se necesita mostrar en el Cashflow es Saldo FF. En este ejemplo 32.700,00



Store procedure creado en la base de factu_full_central

```
use factu_full_central
EXEC cashflow_fondofijo_consulta
```

```
ALTER procedure [dbo].[cashflow_fondofijo_consulta]
(
```

```

        @fecha_ini datetime = null,
        @fecha_fin datetime = null
    )
AS
    DECLARE @fecha_actual DATETIME
    SET @fecha_actual = GETDATE()

CREATE TABLE #general
(
    codi_banco varchar(140),
    cuenta_num varchar(140),
    mone varchar(140),
    impor varchar(140),
    valo_fecha varchar(140),
    valo_num varchar(40),
    valo_tipo varchar(140),
    valo_descrip varchar(140),
    valo_banco_ori varchar(140),
    compro_tipo varchar(140),
    compro_num varchar(140),
    compro_modu varchar(140),
    compro_descrip varchar(140),
    clien_prove varchar(140),
    fecha_real varchar(140),
    obser varchar(140),
    saldo_ini varchar(140),
    descrip_banco varchar(140),
    fecha_alta varchar(140)
)
)

insert into #general (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num, valo_tipo,
valo_descrip, valo_banco_ori, compro_tipo, compro_num, compro_modu, compro_descrip, clien_prove,
fecha_real, obser, saldo_ini, descrip_banco, fecha_alta)
EXEC factu_saldos_banca_lista @fecha_desde = @fecha_actual, @fecha_hasta = @fecha_actual, @codi_banco
= 'FF', @cuenta_num = 'FF Dolares'

select
    g.cuenta_num FONDO,
    CAST(g.saldo_ini AS DECIMAL(10,2)) SALDO_FF
    from #general g

DROP TABLE #general
GO

```

- Consulta al store factu_saldos_banca_lista

```

EXEC factu_saldos_banca_lista @fecha_desde = @fecha_actual, @fecha_hasta = @fecha_actual, @codi_banco =
'FF', @cuenta_num = 'FF Dolares'

```

- Y lo ejecuta en base a la fecha actual. Es decir, mostrará el saldo final en base a la fecha del día de hoy.

```
DECLARE @fecha_actual DATETIME  
SET @fecha_actual = GETDATE()
```

SQL Cheques en cartera

Extracción desde el factu

1. Oprimir VC
2. Estado: En Cartera
3. Oprimir el icono "Buscar"

The screenshot shows the Factu software interface with the following details:

- Toolbar:** Includes icons for various functions like VC, FC, R, A, NC, P, RC, HR, ST, and several others.
- Search Bar:** Contains fields for 'Texto a Buscar...' and a magnifying glass icon.
- Filter Panel:** Shows dropdowns for 'Cliente', 'F. Desde', 'Hasta', 'Vendedor', 'Ubicacion', 'Banco', and 'Estado' (which is set to 'En Cartera').
- Search Button:** A magnifying glass icon with a red box around it, located in the top right of the filter panel.
- Data Grid:** A large table displaying a list of cheques. The columns include:
 - Estado: En Cartera
 - Descripción: Cheque Echeq
 - Nro.: Various values (e.g., 19300101, 03952468, 04042116, etc.)
 - Importe: Various amounts (e.g., 3,027,297.42, 775,156.20, 514,871.00, etc.)
 - Moneda: PES
 - Fecha: Dates ranging from 09/02/2024 to 01/03/2024
 - Banco: Comafi, Provincia Bs As, Galicia, Frances, etc.
 - No a la Orden
 - Int.

CHEQUES en cartera.

Copia esa grilla en el excel y los divide por semana. Cuando ya están cancelados, les cambia el estado y los elimina del cash.

Store procedure creado: **cheques_terce_consul_cashflow** (factu_full_central)

(Se toma de ejemplo el real: **factu_cheques_terce_consul**)

```

use factu_full_central
EXEC cashflow_cheques_terce_consul @estado='C'

```

En Excel

0 ☐ atrasado

6 ☐ semanas siguientes a la actual

1,2,3,4,5 ☐ el valor va depender de la semana actual (columna del centro)

Cheque cartera - Base Central					
Suma de IMPORTE	Etiquetas de columna		0	5	6 Total general
Etiquetas de fila					
■ Banco BPN				\$ 4,000,000.00	\$ 4,000,000.00
■ Carrera Pablo Maximiliano				\$ 4,000,000.00	\$ 4,000,000.00
17/03/2024				\$ 4,000,000.00	\$ 4,000,000.00
■ Comafi	\$	3,027,297.42			\$ 3,027,297.42
■ Sky 222 S.R.L	\$	3,027,297.42			\$ 3,027,297.42
09/02/2024	\$	3,027,297.42			\$ 3,027,297.42
■ Provincia Bs As			\$ 2,014,871.00		\$ 2,014,871.00
■ Simoncini Matias Javier			\$ 514,871.00		\$ 514,871.00
28/02/2024			\$ 514,871.00		\$ 514,871.00
■ NEGRIN LUCAS DAMIAN			\$ 1,500,000.00		\$ 1,500,000.00
28/02/2024			\$ 1,500,000.00		\$ 1,500,000.00
■ ICBC			\$ 800,000.00		\$ 800,000.00
■ MIABUD			\$ 800,000.00		\$ 800,000.00
29/02/2024			\$ 800,000.00		\$ 800,000.00
Total general	\$	3,027,297.42	\$ 2,814,871.00	\$ 4,000,000.00	\$ 9,842,168.42

Versión 1

En Excel

1. Filtrar por mes y del mes se diferencia por número de la semana
2. En las columnas se diferencia por banco e indica la fecha

Cheque cartera - Base Central						
MES	(Todas)					
Suma de IMPORTE	Etiquetas de columna	2	4	5	Total general	
Etiquetas de fila						
■ Banco BPN		\$ 4,000,000.00		\$ 4,000,000.00		
■ Carrera Pablo Maximiliano		\$ 4,000,000.00		\$ 4,000,000.00		
17/03/2024		\$ 4,000,000.00		\$ 4,000,000.00		
■ Comafi	\$ 3,027,297.42			\$ 3,027,297.42		
■ Sky 222 S.R.L	\$ 3,027,297.42			\$ 3,027,297.42		
09/02/2024	\$ 3,027,297.42			\$ 3,027,297.42		
■ Provincia Bs As		\$ 514,871.00	\$ 514,871.00			
■ Simoncini Matias Javier		\$ 514,871.00	\$ 514,871.00			
28/02/2024		\$ 514,871.00	\$ 514,871.00			
Total general	\$ 3,027,297.42	\$ 4,000,000.00	\$ 514,871.00	\$ 7,542,168.42		

Valores propios

Valores propios: Bancos Galicia y Santander Rio, usa principalmente estos bancos.

Store procedure creado: **cheques_propios_lista_cashflow (factu_full_central)**

(Se toma de ejemplo el real: factu_cheques_propios_lista)

Se usa la consulta

```
USE factu_full_central
EXEC cashflow_cheques_propios_lista @estado = 'E'
```

Se tiene que dividir por semana, teniendo en cuenta la semana completa de Lunes a Domingos

Mejorado con DATEPART (WEEK, fecha) NUM_SEMANA

```
select
    t.num NUM,
    t.razon PROVEEDOR,
    p.valo_fecha FECHA,
    abs(p.importe) IMPORTE,
    DATENAME(MONTH, p.valo_fecha) MES, --Para evaluar en qué semana del mes se esta
    (DATEPART(WEEK, p.valo_fecha) - DATEPART(WEEK, DATEADD(DAY, 1 - DATEPART(DAY, p.valo_fecha),
    p.valo_fecha))) + 1 AS NUM_SEMANA_MES,
    --Para evaluar en qué semana del mes actual se esta
    p.mone MONEDA,
    t.estad ESTADO,
    p.cuenta_num CUENTA_ORDEN
from #tempo t, pagos_prove p, valores v
where t.modu = 'P'
    and t.tipo = p.tipo
    and t.num = p.num
    and t.secuen = p.secuen_pago
    and p.valo_tipo = v.codig
    and isnull(p.cuenta_num, '') <> ''
    and (p.importe >= @impor_desde or @impor_desde is null)
    and (p.importe <= @impor_hasta or @impor_hasta is null)

union

select
    t.num NUM,
    t.razon PROVEEDOR,
```

```

i.valo_fecha FECHA,
abs(i.impor) IMPORTE,
DATENAME(MONTH, i.valo_fecha) MES,
(DATEPART(WEEK, i.valo_fecha) - DATEPART(WEEK, DATEADD(DAY, 1 - DATEPART(DAY, i.valo_fecha),
i.valo_fecha))) + 1 AS NUM_SEMANA_MES,
ci.mone MONEDA,
t.está ESTADO,
i.cuenta_num CUENTA_ORDEN
from #tempo t, fon_mov_items i, valores v, fon_mov_cabe ci
where t.modu = 'F'
and t.tipo = i.tipo
and t.num = i.num
and t.secuen = i.secuen
and i.valo_tipo = v.codí
and i.tipo = ci.tipo
and i.num = ci.num
and isnull(i.cuenta_num, '') <> ''
and (i.impor >= @impor_desde or @impor_desde is null)
and (i.impor <= @impor_hasta or @impor_hasta is null)

order by mone,cuenta_orden,t.está

```

Ver full_powerbi_kpi_empresa_cobros

Versión 1

En Excel, pueden filtrar por mes y les indica el número de la semana del mes indicado, y el total por semana

MES	February	▼	
Suma de IMPORTE	NUM_SEMANA_MES	▼	
PROVEEDOR	2	4	5 Total general
BILOTTA GUIDO MAXIMILIANO		\$ 343,091.69	\$ 343,091.69
Coca-Cola FEMSA de Buenos Aires S A	\$ 6,923,657.60	\$ 5,330,823.00	\$ 12,254,480.60
Fratelli Branca Destilerias S.A.	\$ 13,381,041.15		\$ 13,381,041.15
GARCIA DE QUEVEDO LEANDRO HIPOLITO	\$ 480,225.00		\$ 480,225.00
Total general	\$ 20,784,923.75	\$ 5,330,823.00	\$ 343,091.69 \$ 26,458,838.44

Cashflow final

Store procedure creado:

```
ALTER procedure [dbo].[cashflow]
(
    @fecha_desde datetime = null,
    @num_semana varchar (120) = null,
)
as

CREATE TABLE #valor_propio_central
(
    num varchar(20),
    proveedor varchar(50),
    fecha datetime,
    importe numeric(16, 4), /*-----*/
    valo_fecha datetime,
    num_semana varchar(10),
    mes varchar (30),
    moneda varchar(30),
    estado varchar (10),
    cuenta varchar (20)
)

insert into #valor_propio_central(num, proveedor, fecha, importe, valo_fecha, num_semana, mes,
moneda, estado, cuenta)

EXEC factu_full_central.dbo.cashflow_cheques_propios_lista @estado = 'E'

CREATE TABLE #valor_propio_dc
(
    num varchar(20),
    proveedor varchar(50),
    fecha datetime,
    importe varchar(60),
    num_semana varchar(10),
    mes varchar (30),
    moneda varchar(30),
    estado varchar (10),
    cuenta varchar (20)
)

insert into #valor_propio_dc(num, proveedor, fecha, importe, num_semana, mes, moneda, estado, cuenta)
```

```

EXEC factu_full_dist_central.dbo.cashflow_cheques_propios_lista @estado = 'E'

--Tablas temporales creadas caja:
CREATE TABLE #general
(
    codi_banco varchar (20),
    cuenta_num varchar (20),
    mone varchar (20),
    impor varchar (20),
    valo_fecha datetime,
    valo_num varchar (20),
    valo_tipo varchar (20),
    valo_descrip varchar (20),
    valo_banco_ori varchar (20),
    compro_tipo varchar (20),
    compro_num varchar (20),
    compro_modu varchar (20),
    compro_descrip varchar (20),
    clien_prove varchar (20),
    fecha_real varchar (20),
    obser varchar (20),
    saldo_ini varchar (20),
    descrip_banco varchar (20),
    fecha_alta varchar (20)
)

CREATE TABLE #general2
(
    codi_banco varchar (20),
    cuenta_num varchar (20),
    mone varchar (20),
    impor varchar (20),
    valo_fecha datetime,
    valo_num varchar (20),
    valo_tipo varchar (20),
    valo_descrip varchar (20),
    valo_banco_ori varchar (20),
    compro_tipo varchar (20),
    compro_num varchar (20),
    compro_modu varchar (20),
    compro_descrip varchar (20),
    clien_prove varchar (20),
    fecha_real varchar (20),
    obser varchar (20),
    saldo_ini varchar (20),
    descrip_banco varchar (20),
    fecha_alta varchar (20)
)

/*
-----*/

```

```

--Tener en cuenta día actual
DECLARE @fecha_actual DATETIME
SET @fecha_actual = GETDATE()
/*-----*/
insert into #general (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num,
                     valo_tipo, valo_descrip, valo_banco_ori, compro_tipo, compro_num,
                     compro_modu, compro_descrip, clien_prove, fecha_real, obser, saldo_ini,
                     descrip_banco, fecha_alta)

EXEC factu_full_central.dbo.factu_saldos_banca_lista @fecha_desde = @fecha_actual /*de la variable
creada*/, @fecha_hasta = @fecha_actual, @codi_banco = 'CAJ', --@CUENTA_NUM = 'CAJA FLORESTA',
@FECHA_TIPO = 'S', @agregar_mov_auxi = 'N', @cuenta_tipo = 'T', @agrupar_compro = 'N',
@es_admin = 'S', @usuario = 'full_sa', @cod_grupo = 'NULL'

insert into #general2 (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num,
                     valo_tipo, valo_descrip, valo_banco_ori, compro_tipo, compro_num,
                     compro_modu, compro_descrip, clien_prove, fecha_real, obser, saldo_ini,
                     descrip_banco, fecha_alta)

EXEC factu_full_dist_central.dbo.factu_saldos_banca_lista @fecha_desde = @fecha_actual/*de la
variable creada*/, @fecha_hasta = @fecha_actual, @codi_banco = 'CAJ', ---@CUENTA_NUM = 'CAJA FLORESTA',
@FECHA_TIPO = 'S', @agregar_mov_auxi = 'N', @cuenta_tipo = 'T', @agrupar_compro = 'N',
@es_admin = 'S', @usuario = 'full_sa'--, @cod_grupo = 'NULL'

create table #proveedor_central
(
    num varchar(120),
    proveedor varchar(130),
    cod varchar(110),
    comprobante varchar(140),
    num_comp varchar(120),
    sucur varchar(120),
    estado varchar(120),
    agrupacion_prove varchar(120),
    fecha_pago datetime,
    numerosemana varchar(120),
    mes varchar(120),
    total_comp varchar(120),
    suma_total varchar(120),
)
insert into #proveedor_central (num, proveedor, cod, comprobante, num_comp, sucur, estado,
agrupacion_prove, fecha_pago, numerosemana, mes, total_comp, suma_total)
EXEC factu_full_central.dbo.cashflow_proveedor_consulta

create table #proveedor_dc
(
    num varchar(120),
    proveedor varchar(130),

```

```

cod varchar(110),
comprobante varchar(140),
num_comp varchar(120),
sucur varchar(120),
estado varchar(120),
agrupacion_prove varchar(120),
fecha_pago datetime,
numerosemana varchar(120),
mes varchar(120),
total_comp varchar(120),
suma_total varchar(120),
)

insert into #proveedor_dc (num, proveedor, cod, comprobante, num_comp, sucur, estado,
agrupacion_prove, fecha_pago, numerosemana, mes, total_comp, suma_total)
EXEC factu_full_dist_central.dbo.cashflow_proveedor_consulta

create table #cheque_central
(
    tipo varchar(120),
    cod_banco varchar(120),
    banco varchar(120),
    razon varchar(120),
    valo_fecha varchar(120),
    numerosemana varchar(120),
    mes varchar(120),
    importe varchar(120),
    descripcion varchar(120),
    estado varchar(120),
    moneda varchar(120),
    valo_tipo varchar(120),
    secuencia varchar(120)
)

insert into #cheque_central (tipo, cod_banco, banco, razon, valo_fecha, numerosemana, mes, importe,
descripcion, estado, moneda, valo_tipo, secuencia)
EXEC factu_full_central.dbo.cashflow_cheques_terce_consul @estado='C'

CREATE TABLE #fondo_fijo
(
    codi_banco varchar(140),
    cuenta_num varchar(140),
    mone varchar(140),
    impor varchar(140),
    valo_fecha varchar(140),
    valo_num varchar(40),
    valo_tipo varchar(140),
    valo_descrip varchar(140),
    valo_banco_ori varchar(140),

```

```

compro_tipo varchar(140),
compro_num varchar(140),
compro_modu varchar(140),
compro_descrip varchar(140),
clien_prove varchar(140),
fecha_real varchar(140),
obser varchar(140),
saldo_ini varchar(140),
descrip_banco varchar(140),
fecha_alta varchar(140)
)

insert into #fondo_fijo (codi_banco, cuenta_num, mone, impor, valo_fecha, valo_num, valo_tipo,
valo_descrip, valo_banco_ori, compro_tipo, compro_num, compro_modu, compro_descrip, clien_prove,
fecha_real, obser, saldo_ini, descrip_banco, fecha_alta)
EXEC factu_saldos_banca_lista @fecha_desde = @fecha_actual, @fecha_hasta = @fecha_actual, @codi_banco
= 'FF', @cuenta_num = 'FF Dolares'

if @num_semana = datepart(week, getdate())
begin
    SELECT
        '1' as num,
        'FK ' + g.cuenta_num as DESCRIPCION,
        convert(date, getdate()) as fecha, '0' as importe, '-' as num_semana,
        CAST(saldo_ini AS decimal(18,2)) + CAST(impor AS decimal(18,2)) AS SALDO_FF
    FROM #general g
    WHERE g.cuenta_num like 'CAJA%' and g.cuenta_num != 'CAJA MALABIA' and g.cuenta_num != 'CAJA PES
ITURRI'
union all

SELECT
    '1' as num,
    'DC ' + g2.cuenta_num as DESCRIPCION,
    convert(date, getdate()) as fecha, '0' as importe, '-' as num_semana, /*-----*/
    CAST(saldo_ini AS decimal(18,2)) + CAST(impor AS decimal(18,2)) AS SALDO_FF
FROM #general2 g2
WHERE g2.cuenta_num like 'CAJA%' and g2.cuenta_num != 'Caja Dolares' and g2.cuenta_num != 'Caja
Floresta'
union all

select num, 'VP - CENTRAL ' + proveedor as DESCRIPCION, fecha, importe as importe, cast(num_semana
as int) as num_semana, '0' as saldo_ff from #valor_propio_central where num_semana = @num_semana

union all

select num, 'VP - DC ' + proveedor as DESCRIPCION, fecha, importe as importe, cast(num_semana as

```

```

int) as num_semana, '0' as saldo_ff from #valor_propio_dc where num_semana = @num_semana
union all

select num, 'F - CENTRAL ' + proveedor as DESCRIPCION, fecha_pago, total_comp as importe,
cast(numerosemana as int) as num_semana, '0' as saldo_ff from #proveedor_central where numerosemana =
@num_semana
union all

select num, 'F - DC ' + proveedor as DESCRIPCION, fecha_pago, CONVERT(varchar, total_comp) as
importe, cast(numerosemana as int) as num_semana, '0' as saldo_ff from #proveedor_dc where numerosemana =
@num_semana

union all

select '1' as num, 'CHEQUE ' + razon as DESCRIPCION, valo_fecha, CONVERT(varchar, importe ) as
importe, cast(numerosemana as int) as num_semana, '0' as saldo_ff from #cheque_central where
numerosemana = @num_semana

union all

select '1' as num, cuenta_num as DESCRIPCION, convert(date,GETDATE()) as valo_fecha, '0' as importe,
'-' as num_semana, CAST(saldo_ini AS DECIMAL(10,2)) as saldo_ff from #fondo_fijo

order by num_semana
end

else --Que traiga las cajas en 0, si no es la semana actual y fondo fijo también en 0
begin
SELECT
    '1' as num,
    'FK ' + g.cuenta_num as DESCRIPCION,
    convert(date, getdate()) as fecha, '0' as importe, '-' as num_semana,
    '0' AS SALDO_FF

FROM #general g
WHERE g.cuenta_num like 'CAJA%' and g.cuenta_num != 'CAJA MALABIA' and g.cuenta_num != 'CAJA PES
ITURRI'

union all

SELECT
    '1' as num,
    'DC ' + g2.cuenta_num as DESCRIPCION,
    convert(date, getdate()) as fecha, '0' as importe, '-' as num_semana, /*-----*/
    '0' AS SALDO_FF
FROM #general2 g2
WHERE g2.cuenta_num like 'CAJA%' and g2.cuenta_num != 'Caja Dolares' and g2.cuenta_num != 'Caja
Floresta'

union all

```

```

    select num, 'VP - CENTRAL ' + proveedor as DESCRIPCION, fecha, importe as importe, cast(num_semana
as int) as num_semana, '0' as saldo_ff from #valor_propio_central where num_semana = @num_semana

union all

    select num, 'VP - DC ' + proveedor as DESCRIPCION, fecha, importe as importe, cast(num_semana as
int) as num_semana, '0' as saldo_ff from #valor_propio_dc where num_semana = @num_semana
union all

    select num, 'F - CENTRAL ' + proveedor as DESCRIPCION, fecha_pago, total_comp as importe,
cast(numerosemana as int) as num_semana, '0' as saldo_ff from #proveedor_central where numerosemana =
@num_semana
union all

    select num, 'F - DC ' + proveedor as DESCRIPCION, fecha_pago, CONVERT(varchar, total_comp) as
importe, cast(numerosemana as int) as num_semana, '0' as saldo_ff from #proveedor_dc where numerosemana
= @num_semana

union all

    select '1' as num, 'CHEQUE ' + razon as DESCRIPCION, valo_fecha, CONVERT(varchar, importe ) as
importe, cast(numerosemana as int) as num_semana, '0' as saldo_ff from #cheque_central where
numerosemana = @num_semana

union all

    select '1' as num, cuenta_num as DESCRIPCION, convert(date, GETDATE()) as valo_fecha, '0' as importe,
'-' as num_semana, /*CAST(saldo_ini AS DECIMAL(10,2))*/ as saldo_ff from #fondo_fijo

order by num_semana

end

DROP TABLE #valor_propio_central
DROP TABLE #valor_propio_dc
DROP TABLE #general
DROP TABLE #general2
DROP TABLE #proveedor_central
DROP TABLE #proveedor_dc
DROP TABLE #cheque_central
DROP TABLE #fondo_fijo

```

GO

Restricciones:

- Si la semana es igual a la actual. Trae el valor de las cajas y el fondo fijo. Pero, si no, trae el valor de las cajas y el fondo fijo en 0.

```
if @num_semana = datepart(week, getdate())
```

Esto es porque, en el cashflow necesitan solo ver el valor de las cajas y el fondo en la semana actual.

- Se trae los cheques propios en estado 'E' de DC y central

```
EXEC factu_full_central.dbo.cashflow_cheques_propios_lista @estado = 'E'  
EXEC factu_full_dist_central.dbo.cashflow_cheques_propios_lista @estado = 'E'
```

Consulta, ejemplo:

```
exec cashflow @num_semana = '16'
```

Se va consultando por semana en Excel.

Num semana 15			Num semana 16		
	Inicio	Fin		Inicio	Fin
FK CAJA FLORESTA	\$ 797,544.38		FK CAJA FLORESTA	0	
FK Caja Las Heras	\$ 364,433.70		FK Caja Las Heras	0	
FK Caja Monserrat	\$ 297,162.15		FK Caja Monserrat	0	
FK CAJA PES INTERN	\$ 2,337,040.97		FK CAJA PES INTERN	0	
FK Caja Pilar	\$ 599,225.53		FK Caja Pilar	0	
DC Caja Belgrano	\$ 1,192,520.98		DC Caja Belgrano	0	
DC Caja Central	\$ 7,555,470.21		DC Caja Central	0	
DC Caja Cerviño	\$ 2,511,132.05		DC Caja Cerviño	0	
DC Caja Congreso	\$ 883,950.94		DC Caja Congreso	0	
DC Caja Malabia	\$ 2,015,467.34		DC Caja Malabia	0	
DC Caja P.Italia	\$ 4,322,904.67		DC Caja P.Italia	0	
FF Dolares	\$ 6,160.00		FF Dolares	0	

	Fecha	Importe		Fecha	Importe
VP - CENTRAL Fratelli Branca Destilerias S.A.	08/04/2024	\$ 5,969,300.98	VP - CENTRAL La Yungeré S.A.	19/04/2024	\$ 2,149,918.82
VP - CENTRAL Nomina - Cristhian Da Silva Guzi	09/04/2024	\$ 30,308.00	VP - CENTRAL Phaedrus SRL	18/04/2024	\$ 578,791.07
VP - CENTRAL Nomina - Cristhian Da Silva Guzi	09/04/2024	\$ 40,000.00	VP - DC Cermesoni Pablo Hernan	15/04/2024	\$ 1,592,691.71

... Cashflow | S15 | S16 | S17 | S18 | S19 | S20 | S21 | S22 | S23 | S24 | S25 ... + : viernes, 12 de abril de 2024

Intrasistemas - RP

Tabla - INTRA_BRANCH

Campo	Descrip	Ej
sucursal_id	Deposito de la sucursal	BEL
sucursal_nombre	Nombre de la sucursal	Fullescabio Belgrano
sucursal_telefono	Telefono de la sucursal	NULL
sucursal_email	Mail de la sucursal	belgrano@fullescabio.com
sucursal_contacto	Contacto de la sucursal	NULL
sucursal_comentarios	Comentarios sobre la sucursal	NULL
sucursal_direccion	Direccion de la sucursal	Echeverría 2485
sucursal_deposito	Codigo de posito de la sucursal	BEL
sucursal_peya_id	ID Peya de la sucursal	351636
sucursal_rappi_id	ID principal de Rappi	NULL
sucursal_rappi_sub_id	ID secundario de Rappi	NULL

Dentro de esta tabla agregamos la informacion de las sucursales para enviarlas a Intrasistema en con el siguiente SP

SP -

Tabla - INTRASISTEMA_LOC

Dentro de esta Tabla se ingresan las variables que va a utilizar el sp de alta de pedidos . Estas variables definen como se va a crear el pedido, si uno de los datos no esta expresado en esta tabla el pedido no se crea.

Campo	Descripcion	Ej
cliente	Numero de cliente que usa el pedido	11
id_negocio	Id de negocio de Intrasistema	22
tipo	Tipo de pedido a usar	DMA
lista_codi	Lista de precio a usar	MALA
plataforma	Plataforma dentro de Intrasistema	MERCADOLIBRE
agr_2	Agrupacion 2 del cliente	
agr_3	Agrupacion 3 del cliente	
codi_vende	Codigo de vendedor a usar	
depo	Codigo de deposito para reserva	DEP
fc_b_default	FC B default para el cliente	FBC
fc_a_default	FC A default para el cliente	FAC
base	Base donde ingresa el pedido	Central

Vista - INTRASISTEMA_LOC_AGR

Desde esta vista podemos agregar variaciones de descripción de provincia que sean diferentes entre Intrasistema y RP

Campo	Descripción	Ej:
Provincia	Nombre de la provincia desde Intrasistema	Buenos Aires
AGR_1	Agrupacion de la provincia desde RP	BSAS

Vista - INTRASISTEMA_LOC_ESTADO

Esta Vista se usa para transformar los valores de estado entre Intrasistema y RP

Campo	Descripción	Ej:
id_intra	Id de estado Intrasistema	10
descrip_intra	Descripción de Intrasistema	DEVOLUCION- RECIBIDO
cod_rp	Codigo de estado dentro de RP	FAC

SP - intrasistema_branch_list_get

```
ALTER procedure [dbo].[intrasistema_branch_list_get] as  
select * from INTRA_BRANCH
```

Toma los datos de la tabla Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-15794>) para enviarlos hacia intrasistema

SP - intrasistema_invoices

Desde este SP se traen las facturas emitidas a partir de pedidos que hallan integrado con Intrasistema

- Código

```
ALTER procedure [dbo].[intrasistema_invoices]
    @dias int , @bi int
    as
;

if @bi = 21
begin
select  distinct
        c.REMI_NUME id,
        c.fecha fecha_fc,
        c.num nro_fc,
        cc.prefijo nro_pv ,
        c.NUME_CAE nro_cae,
        c.FECHA_CAE fecha_venci_cae ,
        c.ERROR_CAE codigo_cae ,
        c.CUIT_COMPROM cuit_cliente ,
        cc.descrip descrip_compro,
        cc.cod_afip ,
        cc.abreviatura tipo_fc ,
        c.porcen_iva porcen_iva1,
        c.impor_iva impor_iva1,
        c.porcen_iva2 porcen_iva2,
        c.impor_iva2 impor_iva2,
        c.PERCEP_IB_IMPOR + c.IMPOR_IMPUESTS + c.PERCEP_IB2_IMPOR + isnull(c.PERCEP_IVA_IMPOR
,0) otros_impuess,
        c.total,
        c.total - c.impor_iva - c.impor_iva2 - c.PERCEP_IB_IMPOR -
        c.IMPOR_IMPUESTS - c.PERCEP_IB2_IMPOR - isnull(c.PERCEP_IVA_IMPOR ,0)
total_sin_impuess ,
        c.cate_iva,
        cl.num_cliente,
        cl.razon razon_social,
        cl.domicilio ,
        cl.localidad ,
        cl.provincia codi_provincia,
        pr.nombre nombre_provincia ,
```

```

' Impuestos Internos: $' + convert(varchar,convert(numeric(18,2),c.IMPOR_IMPUES)) +
case when isnull(c.PERCEP_IVA_IMPOR,0) != 0 then
' ;Percep. IVA: $' + convert(varchar,convert(numeric(18,2),c.PERCEP_IVA_IMPOR))
else '' end +
case when isnull(c.PERCEP_IB_IMPOR,0) != 0 then
' ;Percep. IB ' + isnull(c.PERCEP_IB_PROVIN,'') + '$' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB_IMPOR)) + '';
else '' end +
case when isnull(c.PERCEP_IB2_IMPOR,0) != 0 then
' ;Percep. IB ' + isnull(c.PERCEP_IB2_PROVIN,'') + '$' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB2_IMPOR )) + '';
else '' end as impuestos ,


c.percep_ib_impor, c.percep_ib_provin ,
c.percep_ib2_impor ,
c.impor_impues_impor_impues_internos ,
c.percep_iva_impor_percep_iva_impor,
c.TIPO_codigo_comprobante,
case when left(c.TIPO,1) = 'F' then concat('Factura ',cc.TIPO) else concat('Nota de
Credito ',cc.TIPO) end as tipo_comprobante,
c.RECI_NUME_id_marketplace,
e.CUIT_emisor_cuit,
e.RAZON_SOCIAL_emisor_razon,
e.CALLE_NUME_emisor_domicilio,
e.INGRE_BRUTOS_iibb ,
e.FECHA_INICIO_ACTIV_fecha_inicio ,
' IVA Responsable Inscripto' condicion_fiscal,
isnull(cl.CP,'S/D') cliente_cp,
case when len(replace(cl.CUIT,'-','')) > 8 then 'CUIT' else 'DNI' end as
cliente_tipo_doc
from comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num=r.num_fact)
inner join comp cc on (c.tipo=cc.cod_comp)
inner join clientes cl on (c.cliente = cl.num_cliente)
left join provincias pr on (cl.provincia = pr.codi)
left join ARTICULOS art on r.ARTICULO = art.COD_ARTICULO
,EMPRESA e

where r.pedi_tipo in ('DEC','DML','dff')
and FECHA >= convert(varchar,getdate() - @dias,23)
and left(c.tipo,1) = 'F'
end

if @bi = 22
begin
select distinct
c.REMI_NUME_id,
c.fecha_fecha_fc,
c.num_nro_fc,
cc.prefijo_nro_pv ,

```

```

c.NUME_CAE nro_cae,
c.FECHA_CAE fecha_venci_cae ,
c.ERROR_CAE codigo_cae ,
c.CUIT_COMPROMIS cuit_cliente ,
cc.descrip descrip_compro,
cc.cod_afip ,
cc.abreviatura tipo_fc ,
c.porcen_iva porcen_iva1,
c.impor_iva impor_iva1,
c.porcen_iva2 porcen_iva2,
c.impor_iva2 impor_iva2,
c.PERCEP_IB_IMPOR + c.IMPOR_IMPUES + c.PERCEP_IB2_IMPOR + isnull(c.PERCEP_IVA_IMPOR
,0) otros_impues,
c.total,
c.total - c.impor_iva - c.impor_iva2 - c.PERCEP_IB_IMPOR -
c.IMPOR_IMPUES - c.PERCEP_IB2_IMPOR - isnull(c.PERCEP_IVA_IMPOR ,0)
total_sin_impues ,
c.cate_iva,
cl.num_cliente,
cl.razon razon_social,
cl.domicilio ,
cl.localidad ,
cl.provincia codi_provincia,
pr.nombre nombre_provincia ,

' Impuestos Internos: $' + convert(varchar,convert(numeric(18,2),c.IMPOR_IMPUES)) +
case when isnull(c.PERCEP_IVA_IMPOR,0) != 0 then
';Percep. IVA: $' + convert(varchar,convert(numeric(18,2),c.PERCEP_IVA_IMPOR))
else '' end +
case when isnull(c.PERCEP_IB_IMPOR,0) != 0 then
';Percep. IB ' + isnull(c.PERCEP_IB_PROVIN,'') + ' $' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB_IMPOR)) + '';
else '' end +
case when isnull(c.PERCEP_IB2_IMPOR,0) != 0 then
';Percep. IB ' + isnull(c.PERCEP_IB2_PROVIN,'') + ' $' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB2_IMPOR )) + '';
else '' end as impuestos ,

c.percep_ib_impor, c.percep_ib_provin ,
c.percep_ib2_impor ,
c.impor_impues impor_impues_internos ,
c.percep_iva_impor percep_iva_impor,
c.TIPO codigo_comprobante,
case when left(c.TIPO,1) = 'F' then concat('Factura ',cc.TIPO) else concat('Nota de
Credito ',cc.TIPO) end as tipo_comprobante,
c.RECI_NUME id_marketplace,
e.CUIT emisor_cuit,
e.RAZON_SOCIAL emisor_razon,
e.CALLE_NUME emisor_domicilio,
e.INGRE_BRUTOS iibb ,

```

```

e.FECHA_INICIO_ACTIV fecha_inicio ,
'IVA Responsable Inscripto' condicion_fiscal,
isnull(cl.CP,'S/D') cliente_cp,
case when len(replace(cl.CUIT,'-','')) > 8 then 'CUIT' else 'DNI' end as
cliente_tipo_doc
from comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num=r.num_fact)
inner join comp cc on (c.tipo=cc.cod_comp)
inner join clientes cl on (c.cliente = cl.num_cliente)
left join provincias pr on (cl.provincia = pr.codi)
left join ARTICULOS art on r.ARTICULO = art.COD_ARTICULO
,EMPRESAS e

where r.pedi_tipo in ('DMA')
and FECHA >= convert(varchar,getdate() - @dias,23)
and left(c.tipo,1) = 'F'
and len(c.REMI_NUME) > 7
end

```

Ejecución

Al ejecutarse el SP se envían la cantidad de días a consultar junto con el business id (código interno de intrasistema que define el negocio/integración)

```
EXEC     intrasistema_invoices @dias = 1, @bi = 21
```

Facturas

```

if @bi = 21
begin
select  distinct
        c.REMI_NUME id,
        c.fecha fecha_fc,
        c.num nro_fc,
        cc.prefijo nro_pv ,
        c.NUME_CAE nro_cae,
        c.FECHA_CAE fecha_venci_cae ,
        c.ERROR_CAE codigo_cae ,
        c.CUIT_COMPROMISARIO cuit_cliente ,
        cc.descrip descrip_compro,
        cc.cod_afip ,
        cc.abreviatura tipo_fc ,
        c.porcen_iva porcen_iva1,
        c.impor_iva impor_iva1,
        c.porcen_iva2 porcen_iva2,
        c.impor_iva2 impor_iva2,

```

```

c.PERCEP_IB_IMPOR + c.IMPOR_IMPUES + c.PERCEP_IB2_IMPOR + isnull(c.PERCEP_IVA_IMPOR ,0)
otros_impues,
c.total,
c.total - c.import_iva - c.import_iva2 - c.PERCEP_IB_IMPOR -
c.IMPOR_IMPUES - c.PERCEP_IB2_IMPOR - isnull(c.PERCEP_IVA_IMPOR ,0) total_sin_impues ,
c.cate_iva,
cl.num_cliente,
cl.razon_razon_social,
cl.domicilio ,
cl.localidad ,
cl.provincia_codi_provincia,
pr.nombre_nombre_provincia ,

' Impuestos Internos: $' + convert(varchar,convert(numeric(18,2),c.IMPOR_IMPUES)) +
case when isnull(c.PERCEP_IVA_IMPOR,0) != 0 then
';Percep. IVA: $' + convert(varchar,convert(numeric(18,2),c.PERCEP_IVA_IMPOR))
else '' end +
case when isnull(c.PERCEP_IB_IMPOR,0) != 0 then
';Percep. IB ' + isnull(c.PERCEP_IB_PROVIN,'') + '$' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB_IMPOR)) + '';
else '' end +
case when isnull(c.PERCEP_IB2_IMPOR,0) != 0 then
';Percep. IB ' + isnull(c.PERCEP_IB2_PROVIN,'') + '$' +
convert(varchar,convert(numeric(18,2),c.PERCEP_IB2_IMPOR )) + '';
else '' end as impuestos ,

c.percep_ib_impor, c.percep_ib_provin ,
c.percep_ib2_impor ,
c.import_impues_import_impues_internos ,
c.percep_iva_impor_percep_iva_impor,
c.TIPO_codigo_comprobante,
case when left(c.TIPO,1) = 'F' then concat('Factura ',cc.TIPO) else concat('Nota de Credito
',cc.TIPO) end as tipo_comprobante,
c.RECI_NUME_id_marketplace,
e.CUIT_emisor_cuit,
e.RAZON_SOCIAL_emisor_razon,
e.CALLE_NUME_emisor_domicilo,
e.INGRE_BRUTOS_iibb ,
e.FECHA_INICIO_ACTIV_fecha_inicio ,
' IVA Responsable Inscripto' condicion_fiscal,
isnull(cl.CP,'S/D') cliente_cp,
case when len(replace(cl.CUIT,'-','')) > 8 then 'CUIT' else 'DNI' end as cliente_tipo_doc

```

Buscamos por tipo de pedido original para ir filtrando los resultados, la variable de @dias se utiliza para buscar la cantidad de días anteriores al día de la fecha de emitida la factura.

Se buscan solo facturas no notas de crédito.

```

from comp_emitidos c inner join reng_fac r on (c.tipo=r.tipo_fact and c.num=r.num_fact)
inner join comp cc on (c.tipo=cc.cod_comp)

```

```
inner join clientes cl on (c.cliente = cl.num_cliente)
left join provincias pr on (cl.provincia = pr.codigoprovincia)
left join ARTICULOS art on r.ARTICULO = art.COD_ARTICULO
,EMPRESA e
where r.pedi_tipo in ('DEC','DML','DFF')
and FECHA >= convert(varchar,getdate() - @dias,23)
and left(c.tipo,1) = 'F'
end
```

SP - intrasistema_items_pricelist_get

Desde este SP informamos la lista de precio hacia Intrasistema, usamos la variable @lista para enviar solo la lista que sea necesaria.

```
ALTER procedure [dbo].[intrasistema_items_pricelist_get]
@lista varchar(5)
as ;
SELECT
LP.LISTA_CODI ,
LP.ARTICULO,
LP.PRECIO_VTA
FROM factu_full_central.dbo.LISTAS_ITEMS AS LP
WHERE lp.LISTA_CODI = @lista AND
(LP.FECHA_HASTA_OFER < SYSDATETIME() OR
 LP.FECHA_HASTA_OFER IS NULL)

UNION ALL
SELECT
LP.LISTA_CODI ,
LP.ARTICULO,
LP.PRECIO_VTA_OFER
FROM factu_full_central.dbo.LISTAS_ITEMS AS LP
where lp.LISTA_CODI = @lista and LP.FECHA_HASTA_OFER > SYSDATETIME()
```

Armado de Lista

La primera sección del código toma los artículos en donde la fecha de fin de oferta halla pasado o este vacía

```
SELECT
LP.LISTA_CODI ,
LP.ARTICULO,
LP.PRECIO_VTA
FROM factu_full_central.dbo.LISTAS_ITEMS AS LP
WHERE lp.LISTA_CODI = @lista AND
(LP.FECHA_HASTA_OFER < SYSDATETIME() OR
 LP.FECHA_HASTA_OFER IS NULL)
```

En la segunda sección tomamos los precios de oferta de los artículos donde la fecha de fin de oferta sea mayor a la fecha del servidor

```
UNION ALL
SELECT
LP.LISTA_CODI ,
LP.ARTICULO,
LP.PRECIO_VTA_OFER
```

```
FROM factu_full_central.dbo.LISTAS_ITEMS AS LP  
where lp.LISTA_CODI = @lista and LP.FECHA_HASTA_OFER > SYSDATETIME()
```

SP - intrasistema_items_stock_get

Desde este SP se envía el stock hacia Intrasistema

•

Codigo

```
ALTER procedure [dbo].[intrasistema_items_stock_get]
    @deposito varchar(5)  as
    set nocount on
    create table #comp_reserva (articulo varchar(60),deposito varchar(4),reservada int)
    create index ind1 on #comp_reserva(articulo);
    create table #art_reserva (articulo varchar(60),deposito varchar(4),reservada int)
    create index ind1 on #art_reserva(articulo);
    create table #art_stock (articulo varchar(60),deposito varchar(4),stock int)
    create index ind1 on #art_stock(articulo);
    if @deposito in ('DEP')
        begin
            insert into #comp_reserva(articulo,deposito,reservada)
            SELECT ac.COD_ARTI_COMPO,p.DEP0_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
                FROM factu_full_central.dbo.PEDI_ITEMS as p
                inner join factu_full_central.dbo.ARTICULOS_COMPO as ac on
                p.ARTICULO=ac.COD_ARTIculo
                WHERE DEPO_RESER = @deposito and p.CANT_PEND <> 0 and RESER = 'S'
                group by ac.COD_ARTI_COMPO,p.DEP0_RESER
            insert into #art_reserva(articulo,deposito,reservada)
            SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA
                FROM factu_full_central.dbo.PEDI_ITEMS
                WHERE RESER = 'S' AND CANT_PEND <> 0 and DEPO_RESER = @deposito GROUP BY ARTICULO,
                DEPO_RESER
            insert into #art_stock(articulo,deposito,stock)
            select cod_articulo,depo,cant_stock from ARTICULOS_STOCK_DEPO where depo = @deposito
            select a.COD_ARTICULO item_id,(isnull(cr.reservada,0) + isnull(ar.reservada,0)) reservado,
            isnull(ars.stock,0) fisico, isnull(ars.stock,0) - (isnull(cr.reservada,0) +
            isnull(ar.reservada,0)) disponible
            from ARTICULOS a
            left join #comp_reserva cr on a.COD_ARTICULO=cr.articulo collate database_default
```

```

left join #art_reserva ar on a.COD_ARTICULO=ar.articulo collate database_default
left join #art_stock ars on a.COD_ARTICULO=ars.articulo collate database_default
where left(a.COD_ARTICULO,1) in ('C','P','G') and left(a.COD_ARTICULO,2) <> 'PR' and a.USA_COMPO =
'N' and a.ACTIVO = 'S'
end

if @deposito = 'BHB'
begin
insert into #comp_reserva(articulo,deposito,reservada)
SELECT ac.COD_ARTI_COMPO,p.DEP0_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
        FROM factu_full_bahia_blanca.dbo.PEDI_ITEMS as p
        inner join factu_full_bahia_blanca.dbo.ARTICULOS_COMPO as ac on
p.ARTICULO=ac.COD_ARTICulo
        WHERE DEPO_RESER = @deposito and p.CANT_PEND <> 0 and RESER = 'S'
        group by ac.COD_ARTI_COMPO,p.DEP0_RESER

insert into #art_reserva(articulo,deposito,reservada)
        SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA
        FROM factu_full_bahia_blanca.dbo.PEDI_ITEMS
        WHERE RESER = 'S' AND CANT_PEND <> 0 and DEPO_RESER = @deposito GROUP BY ARTICULO,
DEPO_RESER

insert into #art_stock(articulo,deposito,stock)
select cod_articulo,depo,cant_stock from factu_full_bahia_blanca.dbo.ARTICULOS_STOCK_DEPO where
depo = @deposito

select a.COD_ARTICULO item_id,(isnull(cr.reservada,0) + isnull(ar.reservada,0)) reservado,
isnull(ars.stock,0) fisico, isnull(ars.stock,0) - (isnull(cr.reservada,0) +
isnull(ar.reservada,0)) disponible
from factu_full_bahia_blanca.dbo.ARTICULOS a
left join #comp_reserva cr on a.COD_ARTICULO=cr.articulo collate database_default
left join #art_reserva ar on a.COD_ARTICULO=ar.articulo collate database_default
left join #art_stock ars on a.COD_ARTICULO=ars.articulo collate database_default
where left(a.COD_ARTICULO,1) in ('C','P','G') and left(a.COD_ARTICULO,2) <> 'PR' and a.USA_COMPO =
'N' and a.ACTIVO = 'S'
end

if @deposito not in ('DEP','BHB')
begin
insert into #comp_reserva(articulo,deposito,reservada)
SELECT ac.COD_ARTI_COMPO,p.DEP0_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
        FROM factu_full_locales.dbo.PEDI_ITEMS as p
        inner join factu_full_locales.dbo.ARTICULOS_COMPO as ac on
p.ARTICULO=ac.COD_ARTICulo
        WHERE DEPO_RESER = @deposito and p.CANT_PEND <> 0 and RESER = 'S'
        group by ac.COD_ARTI_COMPO,p.DEP0_RESER

insert into #art_reserva(articulo,deposito,reservada)
        SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA
        FROM factu_full_locales.dbo.PEDI_ITEMS

```

```

WHERE RESER = 'S' AND CANT_PEND <> 0 and DEPO_RESER = @deposito GROUP BY ARTICULO,
DEPO_RESER

insert into #art_stock(articulo,deposito,stock)
select cod_articulo,depo,cant_stock from factu_full_locales.dbo.ARTICULOS_STOCK_DEPO where depo =
@deposito

select a.COD_ARTICULO item_id,(isnull(cr.reservada,0) + isnull(ar.reservada,0)) reservado,
isnull(ars.stock,0) fisico, isnull(ars.stock,0) - (isnull(cr.reservada,0) +
isnull(ar.reservada,0)) disponible
from factu_full_locales.dbo.ARTICULOS a
left join #comp_reserva cr on a.COD_ARTICULO=cr.articulo collate database_default
left join #art_reserva ar on a.COD_ARTICULO=ar.articulo collate database_default
left join #art_stock ars on a.COD_ARTICULO=ars.articulo collate database_default
where left(a.COD_ARTICULO,1) in ('C','P','G') and left(a.COD_ARTICULO,2) <> 'PR' and a.USA_COMPO =
'N' and a.ACTIVO = 'S'
end

```

Creación de Tablas

Creamos las tablas temporales donde vamos a insertar las reservas

```

create table #comp_reserva (articulo varchar(60),deposito varchar(4),reservada int)
create index ind1 on #comp_reserva(articulo);

create table #art_reserva (articulo varchar(60),deposito varchar(4),reservada int)
create index ind1 on #art_reserva(articulo);

create table #art_stock (articulo varchar(60),deposito varchar(4),stock int)
create index ind1 on #art_stock(articulo);

```

Stock

Dependiendo de la variable @depo vamos a informar el stock de diferentes bases

```

if @deposito in ('DEP')

begin

```

Cargamos dentro de la tabla comp_reserva los articulos compuestos que tengan reserva. Por ejemplo si un articulo compuesto esta conformado por una coca y un fernet y se hace una reserva sobre este articulo se agrega una coca y un fernet.

```

insert into #comp_reserva(articulo,deposito,reservada)
SELECT ac.COD_ARTI_COMPO,p.DEP0_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
FROM factu_full_central.dbo.PEDI_ITEMS as p
inner join factu_full_central.dbo.ARTICULOS_COMPO as ac on p.ARTICULO=ac.COD_ARTICULO
WHERE DEPO_RESER = @deposito and p.CANT_PEND <> 0 and RESER = 'S'
group by ac.COD_ARTI_COMPO,p.DEP0_RESER

```

Cargamos en la tabla art_reservar los articulos que estén reservando stock

```
insert into #art_reserva(articulo,deposito,reservada)
    SELECT ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA
    FROM factu_full_central.dbo.PEDI_ITEMS
    WHERE RESER = 'S' AND CANT_PEND <> 0 and DEPO_RESER = @deposito GROUP BY ARTICULO,
DEPO_RESER
```

Cargamos el stock actual de los articulos

```
insert into #art_stock(articulo,deposito,stock)
select cod_articulo,depo,cant_stock from ARTICULOS_STOCK_DEPO where depo = @deposito
```

Sumamos las cantidades reservadas de la tabla art_reserva y comp_reservar para restarla con la tabla de art_stock para tener como resultado el stock actual.

Solo traemos los articulos que estén activos y empiecen con C (articulo normal), P (articulo promocional), G (articulo Malaria). Se excluyen los articulos PR (promociones) y los articulos que tengan componentes.

```
select a.COD_ARTICULO item_id,(isnull(cr.reservada,0) + isnull(ar.reservada,0)) reservado,
isnull(ars.stock,0) fisico, isnull(ars.stock,0) - (isnull(cr.reservada,0) + isnull(ar.reservada,0))
disponible
from ARTICULOS a
left join #comp_reserva cr on a.COD_ARTICULO=cr.articulo collate database_default
left join #art_reserva ar on a.COD_ARTICULO=ar.articulo collate database_default
left join #art_stock ars on a.COD_ARTICULO=ars.articulo collate database_default
where left(a.COD_ARTICULO,1) in ('C','P','G') and left(a.COD_ARTICULO,2) <> 'PR' and a.USA_COMPO = 'N'
and a.ACTIVO = 'S'
end
```

SP - intrasistema_items_stock_get_v2

SP - intrasistema_payment_post

.

Codigo SQL

```
@procesador_orden_id varchar(50) , -- numero de pedido intra  
@plataforma varchar(50), -- plataforma de donde salio el pedido  
@plataforma_pago varchar(50) , --nombre de la plataforma de los pagos  
@plataforma_pago_tipo varchar(12) , -- si es credito o debito  
@plataforma_pago_id varchar(50) , -- numero de transaccion de ml  
@plataforma_pago_sub_id varchar(50) , --numero de envio ml  
@plataforma_pago_descripcion varchar(200) , --descripcion del pago  
@plataforma_pago_monto numeric(12,4) , -- monto de la transaccion  
@plataforma_pago_fecha datetime , --fecha de creacion del pago en la plataforma  
@plataforma_pago_id_interno varchar (100),  
@plataforma_pago_tipo_transaccion varchar (100),  
@plataforma_pago_tipo_transaccion_sub varchar(100),  
@plataforma_pago_tipo_transaccion_interno varchar(100),  
@plataforma_orden_id varchar(50) , --orden del pedido que entro el pago  
@plataforma_orden_sub_id varchar(50), -- numero alternativo de la orden (packid en ML)  
@procesador_negocio_id int  
  
as  
-----campos extra-----  
declare  
@cliente int = null,  
@num int = null,  
@total numeric(12,4) = null,  
@check_valores int = null,  
@check_valores_pagos int = null,  
@mone_alternativa numeric(12,4) =null,  
@cat_iva varchar(3) = null,  
@oc_nume varchar(50) = null,  
@obs_1 varchar(50) = null,  
@check_pedido int = null,  
@tipo varchar(3) = null,  
@renglon int = null  
  
-----validaciones-----  
if isnull(@procesador_orden_id,null) is null  
begin  
    select '1' error,'El numero de pedido de INTRA no puede ser NULL' descripcion  
    return  
end  
  
/*  
select @check_pedido = 1 from PEDI_CABE where REFERENCIA = @procesador_orden_id
```

```
if isnull(@procesador_orden_id,0) <> 1
begin
    select '1' error,'El nro de orden Tecnolog Nro.'+@procesador_orden_id+' no se encuentra dentro
de RP' descripcion
    return
end
*/
-----
```

```
select @mone_alternativa = MONE_COTI from MONEDAS where CODI_AFIP = 'DOL'
```

```
if @procesador_negocio_id = '22'
begin
    select 1 error,'Malaria no carga anticipos' descripcion
end
else
begin
begin transaction
if isnull(@procesador_orden_id,'') <> ''
```

```
begin
```

```
if @plataforma_pago_tipo = 'debit' and left(@plataforma_pago_descripcion,1) <> 'A'
or @plataforma_pago_tipo = 'credit' and left(@plataforma_pago_descripcion,1) = 'A'
or @plataforma_pago_tipo = 'neutral' and @plataforma_pago_tipo_transaccion = 'taxis'
```

```
begin
```

```
-----carga de anticipo proveedor-----
```

```
-----cambio de cliente para envio de ml-----
```

```
if @plataforma_pago_descripcion like 'D%envi%'
begin
    set @cliente = '8101'
end
else
begin
    set @cliente = '8100'
end
```

```
----revision para que no se dupliquen los comprobantes
```

```
if @plataforma_pago_tipo = 'credit' and left(@plataforma_pago_descripcion,1) = 'A'
begin
set @tipo = 'NN2'
end
else
begin
```

```

set @tipo = 'NNN'
end

select @num = num from COMPRO_CABE_PROVE
where TIPO = @tipo and PROVE = @cliente and RECI_NUME =
right(isnull(@plataforma_orden_sub_id,@plataforma_orden_id),15)

--begin transaction

if isnull(@num, 0) = 0
begin

select @num = PROX_NUM from COMP where COD_COMP = @tipo
update comp set PROX_NUM = @num +1 where COD_COMP = @tipo

insert into
COMPRO_CABE_PROVE(num, tipo, prove, FECHA, OBSER, RECI_NUME, USER_ALTA, FECHA_ALTA, HOST_NOMBRE, ESTADO, SUCUR, SECUEN, PERCEP_IVA, PERCEP_IB, PORCEN_IVA, PORCEN_IVA2, CATE_IVA, MONE_COTI, IMPOR_IVA, IMPOR_IVA2, MONE, PERCEP_GCIA, SUB_TOTAL_EXEN, OTROS_IMPUESTOS, PORCEN_OTROS_IMPUESTOS, PERCEP_IB2, IMPOR_IVA3, PORCEN_IVA3, IMPOR_IVA4, PORCEN_IVA4, MONE_ALTER_COTI)
select @num ,@tipo,@cliente,convert(varchar,getdate(),23)
,@plataforma_orden_id,right(isnull(@plataforma_orden_sub_id,@plataforma_orden_id),15),substring(user_sname(),1,20),getdate(),substring(host_name(),1,30),'PEN',0,1,0,0,0,0,'RI',1,0,0,'PES',0,0,0,0,0,0,0,0,@mone_alternativa

if @@error<>0
begin
    rollback transaction
    return
end

end

-----control para no cargar el mismo monto dos veces-----
select @check_valores = 1 from PAGOS_PROVE where tipo = @tipo and num = @num and PROVE = @cliente
and IMPORTE = @plataforma_pago_monto
and VALO_NUM = @plataforma_pago_id and VALO_OBSER
= left(@plataforma_pago_descripcion,50)

if isnull(@check_valores,0) = 0

begin
insert into
PAGOS_PROVE(tipo, num, PROVE, FECHA, IMPORTE, VALO_TIPO, VALO_BANCO, CUENTA_NUM, VALO_NUM, VALO_FECHA, MONE_COTI, MONE, RECI_NUME, VALO_OBSER, SUCUR, SECUEN, ESTA, PAGOS_SECUEN, PORCEN_RETEN, BASE_IMPO_RETEN)
select @tipo,@num,@cliente,getdate(),@plataforma_pago_monto,'TR','MLP','Mercado
Pago',isnull(@plataforma_pago_sub_id,@plataforma_pago_id),getdate(),1,'PES',isnull(@plataforma_orden_sub_id,@plataforma_orden_id),left(@plataforma_pago_descripcion,50),0,1,'',0,0,0
if @@error<>0

```

```

begin
    rollback transaction
    return
end

select @total = sum(importe)*-1 from PAGOS_PROVE where tipo = @tipo and num = @num group by
tipo,NUM

update COMPRO_CABE_PROVE set SUB_TOTAL = @total , TOTAL = @total , SALDO = @total where tipo =
@tipo and num = @num

if @@error<>0
    begin
        rollback transaction
        return
    end

end
end

else
-----carga de anticipo cliente -----

begin

if left(@plataforma_pago_descripcion,1) = 'A' and @plataforma_pago_tipo = 'debit'
begin
set @tipo = 'BDM'

select distinct @check_valores = 1 from COMP_EMITIDOS where tipo = @tipo and RECI_NUME =
isnull(@plataforma_orden_sub_id,@plataforma_orden_id)

-----validacion de anticipo ya creado-----
if isnull(@check_valores,0) <> 1

begin
    begin

select @cliente = CLIENTE, @oc_nume = OC_NUME ,@obs_1 = OBS   from pedi_cabe where REFERENCIA =
@procesador_orden_id

select @cat_iva = IVA
      from clientes where NUM_CLIENTE = @cliente
    end

if isnull(@num,0) = 0
    begin

```

```

select @num = PROX_NUM from COMP where COD_COMP = @tipo
update comp set PROX_NUM = @num +1 where COD_COMP = @tipo
if @@error<>0
begin
    rollback transaction
    return
end

insert into
COMP_EMITIDOS(num, tipo, fecha, cliente, estado, VENCIMIENTO, OBSER, porcen_iva, cate_iva, ACTUA_STOCK, MONE
_COTI, TIPO_MOVI,
impor_iva, PORCEN_IVA2, IMPOR_IVA2, PORCEN_DESCUEN, MONE, SUB_TOTAL_EXEN, RECI_NUME, USER_ALTA, FECHA_ALTA
,PORCEN_COMI, NOMBRE_PACIENTE--numero de operacion de ML
,HOST_NOMBRE, OBSER_CONTROL)
select
@num,@tipo,convert(varchar,getdate(),23),isnull(@cliente,8),'PEN',convert(varchar,getdate(),23), '#
'+@oc_numet+' '+@obs_1,0,@cat_iva,'N',1,'I',
0,0,0,0,'PES',0,isnull(@plataforma_orden_sub_id,@plataforma_orden_id),substring(suser_sname(),1,20
),getdate(),0,@plataforma_pago_id
,substring(host_name(),1,30),@plataforma_orden_sub_id
if @@error<>0
begin
    rollback transaction
    return
end
end

select distinct @num = num from COMP_EMITIDOS where tipo = @tipo and reci_nume =
isnull(@plataforma_orden_sub_id,@plataforma_orden_id)

select distinct @check_valores_pagos = 1
from RENG_FAC
where TIPO_FACT = @tipo and NUM_FACT = @num
and TAMANO = @plataforma_pago_id
and COLOR = left(@plataforma_pago_descripcion,50)
and PRECIO = @plataforma_pago_monto

select @renglon = isnull(max(renglon),0) from RENG_FAC where TIPO_FACT = @tipo and NUM_FACT = @num

if isnull(@check_valores_pagos,0) = 0

begin
    insert into
RENG_FAC(TIPO_FACT,NUM_FACT,REGLON,ARTICULO,CANT,PRECIO,COLOR,TAMANO,PRECIO_NETO,UM_CONVERTIR)
select

```

```

@tipo,@num,@renglon+1,'A47',-1,@plataforma_pago_monto, left(@plataforma_pago_descripcion,50),@plataforma_pago_id,@plataforma_pago_monto,1
    if @@error<>0
begin
    rollback transaction
    return
end

select @total = sum(PRECIO) from RENG_FAC where TIPO_FACT = @tipo and NUM_FACT = @num group by
TIPO_FACT,NUM_FACT

update COMP_EMITIDOS set TOTAL = abs(@total),SUB_TOTAL = abs(@total) ,SALDO = abs(@total) where
tipo = @tipo and num = @num
if @@error<>0
begin
    rollback transaction
    return
end

end

end
else
set @tipo = 'A40'

select distinct @check_valores = 1 from COMP_EMITIDOS where tipo = @tipo and RECI_NUME =
isnull(@plataforma_orden_sub_id,@plataforma_orden_id)

-----validacion de anticipo ya creado-----
if isnull(@check_valores,0) <> 1

begin
begin

select @cliente = CLIENTE, @oc_nume = OC_NUME ,@obs_1 = OBS   from pedi_cabe where REFERENCIA =
@procesador_orden_id

select @cat_iva = IVA
      from clientes where NUM_CLIENTE = @cliente
end

if isnull(@num,0) = 0
begin

select @num = PROX_NUM from COMP where COD_COMP = @tipo
update comp set PROX_NUM = @num +1 where COD_COMP = @tipo

```

```

if @@error<>0
begin
    rollback transaction
    return
end

insert into
COMP_EMITIDOS(num, tipo, fecha, cliente, estado, VENCIMIENTO, OBSER, porcen_iva, cate_iva, ACTUA_STOCK, MONE
_COTI, TIPO_MOVI,

impor_iva, PORCEN_IVA2, IMPOR_IVA2, PORCEN_DESCUEN, MONE, SUB_TOTAL_EXEN, RECI_NUME, USER_ALTA, FECHA_ALTA
,PORCEN_COMI, NOMBRE_PACIENTE--numero de operacion de ML
,HOST_NOMBRE, OBSER_CONTROL)
select
@num,@tipo,convert(varchar,getdate(),23),isnull(@cliente,8),'PEN',convert(varchar,getdate(),23),#
'+@oc_numet' '+@obs_1,0,@cat_iva,'N',1,'I',
0,0,0,0,'PES',0,isnull(@plataforma_orden_sub_id,@plataforma_orden_id),substring(suser_sname(),1,20
),getdate(),0,@plataforma_pago_id
,substring(host_name(),1,30),@plataforma_orden_sub_id
if @@error<>0
begin
    rollback transaction
    return
end
end

select distinct @num = num from COMP_EMITIDOS where tipo = @tipo and reci_nume =
isnull(@plataforma_orden_sub_id,@plataforma_orden_id)

select distinct @check_valores_pagos = 1
from pagos
where tipo = @tipo and num = @num
and valo_num = @plataforma_pago_id
and valo_obsr = left(@plataforma_pago_descripcion,50)
and IMPORTE = @plataforma_pago_monto

if isnull(@check_valores_pagos,0) = 0
begin
    insert into
PAGOS(tipo,num,FECHA,IMPORTE,VALO_TIPO,TARJE_CODI,VALO_NUM,VALO_FECHA,MONE,MONE_COTI,ESTA
,CUOTAS,BANCO_DESTI,CUENTA_NUM,PORCEN_RETEN,BASE_IMPO_RETEN,VALO_OBSER)
    select @tipo,@num,convert(varchar,getdate(),23),@plataforma_pago_monto,case when
@plataforma_pago_sub_id is null then 'TR' else 'MP' end,case when @plataforma_pago_sub_id is null
then NULL else 'MLP'
```

```

end, isnull(@plataforma_pago_sub_id,@plataforma_pago_id),convert(varchar,@plataforma_pago_fecha,23)
,'PES',1,'P'
      ,0,'MLP','Mercado Pago',0,0, left(@plataforma_pago_descripcion,50)
      if @@error<>0
begin
      rollback transaction
      return
end

select @total = sum(importe)*-1 from pagos where tipo = @tipo and num = @num group by tipo,NUM

update COMP_EMITIDOS set TOTAL = @total,SUB_TOTAL = @total ,SALDO = @total where tipo = @tipo and
num = @num
if @@error<>0
begin
      rollback transaction
      return
end

end
end
end
commit transaction
select 1 error , 'Comprobante creado '+@tipot+'-'+cast(@num as varchar) descripcion

end

```

Campos para ejecucion

Todos los campos son sacados desde la tabla payments_transactions del servidor de intrasistemas

```

@procesador_orden_id varchar(50) , -- numero de pedido intra
@plataforma varchar(50), -- plataforma de donde salio el pedido
@plataforma_pago varchar(50) , --nombre de la plataforma de los pagos
@plataforma_pago_tipo varchar(12) , -- si es credito o debito
@plataforma_pago_id varchar(50) , -- numero de transaccion de ml
@plataforma_pago_sub_id varchar(50) , --numero de envio ml
@plataforma_pago_descripcion varchar(200) , --descripcion del pago
@plataforma_pago_monto numeric(12,4) , -- monto de la transaccion
@plataforma_pago_fecha datetime , --fecha de creacion del pago en la plataforma
@plataforma_pago_id_interno varchar (100),
@plataforma_pago_tipo_transaccion varchar (100),
@plataforma_pago_tipo_transaccion_sub varchar(100),
@plataforma_pago_tipo_transaccion_interno varchar(100),

```

```
@plataforma_orden_id varchar(50) , --orden del pedido que entro el pago  
@plataforma_orden_sub_id varchar(50), -- numero alternativo de la orden (packid en ML)  
@procesador_negocio_id int -- id del negocio interno de intra
```

Declaracion de variables

Declaramos las variables que vamos a usar dentro del store ya sea para validaciones o valores

```
declare  
@cliente int = null,  
@num int = null,  
@total numeric(12,4) = null,  
@check_valores int = null,  
@check_valores_pagos int = null,  
@mone_alternativa numeric(12,4) =null,  
@cat_iva varchar(3) = null,  
@oc_nume varchar(50) = null,  
@obs_1 varchar(50) = null,  
@check_pedido int = null,  
@tipo varchar(3) = null,  
@renglon int = null
```

Validaciones de Ejecucion

Control de datos para el ingreso de los pedidos, controles activos

- No se puede enviar un valor nulo para el numero de predio de intrasistemas
- No se cargan los anticipos para el negocio 22 (Malaria)

Hay una validación extra comentada, valida si el pedido fue creado dentro del sistema antes de enviarse la informacion del pago. Se deshabilito porque el envio de informacion de pedidos es asincrónico por estos momentos.

- El store consulta principalmente el numero de cliente que tiene cargado el pedido para machearlo con el anticipo
 - Ahora tiene puesto el numero de cliente 8 cuando no encuentra el pedido original

```
if isnull(@procesador_orden_id,null) is null  
begin  
    select '1' error,'El numero de pedido de INTRA no puede ser NULL' descripcion  
    return
```

```

end

/*
select @check_pedido = 1 from PEDI_CABE where REFERENCIA = @procesador_orden_id
if isnull(@procesador_orden_id,0) <> 1
begin
    select '1' error,'El nro de orden Tecnogo Nro.'+@procesador_orden_id+' no se encuentra dentro de RP'
descripcion
    return
end
*/
if @procesador_negocio_id = '22'
begin
    select 1 error,'Malaria no carga anticipos' descripcion
end

```

Carga de comprobantes Proveedor

Validaciones

if @plataforma_pago_tipo = 'debit' and left(@plataforma_pago_descripcion,1) <> 'A'

Si el pedido entra como tipo debito y en la descriucion no es una anulacion

- Carga todos los debitos como impuestos, comiciones , gastos de envios

or @plataforma_pago_tipo = 'credit' and left(@plataforma_pago_descripcion,1) = 'A'

Si el pedido es un credito pero muestra que es una anulacion.

- Cuando se produce una cancelacion ingresamos el reverso como un nuevo tipo de comprobante para no modificar el original

or @plataforma_pago_tipo = 'neutral' and @plataforma_pago_tipo_transaccion = 'taxs'

Si esta marcado como neutral pero la transaccion es de tipo impuesto

- Se cargo de esta manera para los casos donde intrasistemas toma el movimiento como neutral pero dentro de mercado pago el movimiento es de signo negativo

Costo de Envio

Cuando el movimiento es por el costo de envio que mercado libre descuenta de la transaccion se cambia el cliente al 8101 (Melilog SRL)

```

if @plataforma_pago_descripcion like 'D%envi%'
begin
    set @cliente = '8101'
end
else
begin
    set @cliente = '8100'
end

```

Anulaciones

Para las anulaciones de ventas se carga el reverso en otro tipo de comprobante para no modificar el cargado normalmente

```

if @plataforma_pago_tipo = 'credit' and left(@plataforma_pago_descripcion,1) = 'A'
begin
    set @tipo = 'NN2'
end
else
begin
    set @tipo = 'NNN'
end

```

Revisión de Comprobante Duplicado

Se consulta la base buscando los campos de Pack id / Numero de orden para el tipo de comprobante que se esta usando. Si devuelve un numero de comprobante no se ejecuta la seccion de carga de uno nuevo

```

select @num = num from COMPRO_CABE_PROVE
where TIPO = @tipo and PROVE = @cliente and RECI_NUME =
right(isnull(@plataforma_orden_sub_id,@plataforma_orden_id),15)

--begin transaction

if isnull(@num,0) = 0
begin

```

Creacion de comprobante

Consultamos el proximo numero para el comprobante y lo actualisamos. Despues se crea el comprobante usando los datos que nos envia intra, si muestra algun error en la ejecucion se vuelve atras la transacciones. No se guarda

```

select @num = PROX_NUM from COMP where COD_COMP = @tipo
update comp set PROX_NUM = @num +1 where COD_COMP = @tipo

insert into
COMPRO_CABE_PROVE(num, tipo, prove, FECHA, OBSER, RECI_NUME, USER_ALTA, FECHA_ALTA, HOST_NOMBRE, ESTADO, SUCUR, SECU
EN, PERCEP_IVA, PERCEP_IB, PORCEN_IVA, PORCEN_IVA2, CATE_IVA, MONE_COTI, IMPOR_IVA, IMPOR_IVA2, MONE, PERCEP_GCIA, S
UB_TOTAL_EXEN, OTROS_IMPUESTOS, PORCEN_OTROS_IMPUESTOS, PERCEP_IB2, IMPOR_IVA3, PORCEN_IVA3, IMPOR_IVA4, PORCEN_IVA4, M
ONE.Alter_COTI)
select @num ,@tipo,@cliente,convert(varchar,getdate(),23)
,@plataforma_orden_id,right(isnull(@plataforma_orden_sub_id,@plataforma_orden_id),15),substring(suser_sna
me(),1,20),getdate(),substring(host_name(),1,30),'PEN',0,1,0,0,0,0,'RI',1,0,0,'PES',0,0,0,0,0,0,0,0,@mo
ne_alternativa

if @@error<>0
begin
    rollback transaction
    return
end

```

Revisión de valores duplicados

Se revisa que no se cargue el mismo valor dos veces, si se encuentra que un valor es igual a uno cargado anteriormente no se carga

```

select @check_valores = 1 from PAGOS_PROVE where tipo = @tipo and num = @num and PROVE = @cliente and
IMPORTE = @plataforma_pago_monto
                                         and VALO_NUM = @plataforma_pago_id and VALO_OBSER =
left(@plataforma_pago_descripcion,50)

if isnull(@check_valores,0) = 0

```

SP - intrasistema_order_cancel

Este sp se usa para buscar las nuevas

•

Código

```
ALTER procedure [dbo].[intrasistema_order_cancel]

@procesador_order_id varchar(200) ,
@procesador_obs varchar(200)

as

declare @tipo varchar(3),@num int,@cod_envio int,@plataforma_envio_id varchar(200),@obs_3
varchar(200),
@estado varchar(200) = null,
@num_lote int = null,
@LOTE_CONFIRMADO int = null,
@FECHA_INICIO_ARMADO datetime = null,
@USUARIO_ARMADO varchar(30) = null,
@TIPO_FC_PICKING varchar(3) = null,
@NUM_FC_PICKING int = null,
@FECHA_ENTREGA_LOGIS datetime = null,
@USUARIO_ENTREGA varchar(30) = null,
@referencia varchar(30) = null,
@nro_orden varchar(30) = null,
@esta_pedido varchar(4) = null,
@prove int = null,
@total int = null,
@reci_num varchar(200) = null,
@remi_num varchar(200) = null,
@modificado varchar(30) = null

select  @referencia = REFERENCIA,
@num_lote = lote_nume,
@LOTE_CONFIRMADO = LOTE_CONFIRMADO,
@FECHA_INICIO_ARMADO = FECHA_INICIO_ARMADO,
@USUARIO_ARMADO = USUARIO_ARMADO,
@TIPO_FC_PICKING = TIPO_FC_PICKING ,
@NUM_FC_PICKING = NUM_FC_PICKING,
@FECHA_ENTREGA_LOGIS = FECHA_ENTREGA_LOGIS,
@USUARIO_ENTREGA = USUARIO_ENTREGA ,
@esta_pedido = ESTADO,
@tipo = TIPO , @num = NUM ,@prove = CODI_TRANSPOR
```

```

,@plataforma_envio_id = externalShippingId, @obs_3 = concat(OBS_3,' | ',@procesador_obs,' |
Cambio a estado rechazado por Intrasistemas el: ',convert(varchar,getdate(),9)),@modificado = nota
from PEDI_CABE where REFERENCIA = @procesador_order_id

if @modificado is null
begin
    if @esta_pedido = 'CAN' set @estado = 'Cancelado'
    if @num_lote is null set @estado = 'Pendiente'
    if @num_lote is not null and @LOTE_CONFIRMADO is null set @estado = 'Picking'
    if @LOTE_CONFIRMADO = '1' and @FECHA_INICIO_ARMADO is null set @estado = 'Cola_Armado'
    if @FECHA_INICIO_ARMADO is not null set @estado = 'Armando'
    if @TIPO_FC_PICKING is not null set @estado = 'Facturado'
    if @FECHA_ENTREGA_LOGIS is not null set @estado = 'Entregado'

    set @modificado = 'Modf. Proc. Cancelado'

    if @estado in ('Pendiente','Picking','Cola_Armado','Armando')
        begin
            update PEDI_CABE set ESTADO = 'CAN' , OBS_3 = left(@obs_3,2000), ExternalShippingId =
@plataforma_envio_id+'_CAN',nota = @modificado where TIPO = @tipo and num = @num
            update PEDI_ITEMS set CODI_ESTA = 'REC', RESER = 'N',FECHA_CAMBIO_ESTA = getdate()
        where TIPO = @tipo and NUM = @num
        end

    if @estado = 'Facturado'
        begin
            update PEDI_CABE set OBS_3 = left(@obs_3,2000), ExternalShippingId =
@plataforma_envio_id+'_CAN',nota = @modificado where TIPO = @tipo and num = @num
        end

    if @estado = 'Entregado'
        begin
            update PEDI_CABE set OBS_3 = left(@obs_3,2000),nota = @modificado where TIPO = @tipo
and num = @num

            declare @obs varchar(200) = null
            select @reci_num =RECI_NUME, @total = total,@remi_num = REMI_NUME from COMP_EMITIDOS
            where TIPO = @TIPO_FC_PICKING and num = @NUM_FC_PICKING
            select @obs = CONCAT('#IS ',@remi_num,' Factura ',concat(@TIPO_FC_PICKING,' -
',right(concat('00000000',@NUM_FC_PICKING),8)))
        end
    end

select 0 error, 'Pedido modificado'

```

Declaramos las variables

```

declare @tipo varchar(3),@num int,@cod_envio int,@plataforma_envio_id varchar(200),@obs_3
varchar(200),
@estado varchar(200) = null,
@num_lote int = null,
@LOTE_CONFIRMADO int = null,
@FECHA_INICIO_ARMADO datetime = null,
@USUARIO_ARMADO varchar(30) = null,
@TIPO_FC_PICKING varchar(3) = null,
@NUM_FC_PICKING int = null,
@FECHA_ENTREGA_LOGIS datetime = null,
@USUARIO_ENTREGA varchar(30) = null,
@referencia varchar(30) = null,
@nro_orden varchar(30) = null,
@esta_pedido varchar(4) = null,
@prove int = null,
@total int = null,
@reci_num varchar(200) = null,
@remi_num varchar(200) = null,
@modificado varchar(30) = null

```

Selección de Pedido

Ingresamos datos a las variables buscando el numero de pedido de intrasistemas dentro de la base de locales

```

select    @referencia = REFERENCIA,
          @num_lote = lote_nume,
          @LOTE_CONFIRMADO = LOTE_CONFIRMADO,
          @FECHA_INICIO_ARMADO = FECHA_INICIO_ARMADO,
          @USUARIO_ARMADO = USUARIO_ARMADO,
          @TIPO_FC_PICKING = TIPO_FC_PICKING ,
          @NUM_FC_PICKING = NUM_FC_PICKING,
          @FECHA_ENTREGA_LOGIS = FECHA_ENTREGA_LOGIS,
          @USUARIO_ENTREGA = USUARIO_ENTREGA ,
          @esta_pedido = ESTADO,
          @tipo = TIPO , @num = NUM ,@prove = CODI_TRANSPOR
        ,@plataforma_envio_id = externalShippingId, @obs_3 = concat(OBS_3,' | ',@procesador_obs,' |
Cambio a estado rechazado por Intrasistemas el: ',convert(varchar,getdate(),9)),@modificado =
nota
from PEDI_CABE where REFERENCIA = @procesador_order_id

```

Estado

Definimos el estado del pedido dependiendo de las variables

1. Si el estado del pedido es CAN el estado es Cancelado

2. Si no tienen numero de lote el estado es Pendiente
3. Si tiene numero de lote pero no esta confirmado el estado es Picking
4. Si el lote esta confirmado y no tiene inicio de armado el estado es Cola de Armado
5. Si tiene fecha de inicio de armado el estado es Armando
6. Si el tiene un tipo de factura el estado es Facturado
7. Si tiene una fecha de entrega logistica el estado es Entregado

Agregamos el campo @modificado para no duplicar el proceso de cancelacion

```

if @modificado is null
begin
    if @esta_pedido = 'CAN' set @estado = 'Cancelado'
    if @num_lote is null set @estado = 'Pendiente'
    if @num_lote is not null and @LOTE_CONFIRMADO is null set @estado = 'Picking'
    if @LOTE_CONFIRMADO = '1' and @FECHA_INICIO_ARMADO is null set @estado = 'Cola_Armado'
    if @FECHA_INICIO_ARMADO is not null set @estado = 'Armando'
    if @TIPO_FC_PICKING is not null set @estado = 'Facturado'
    if @FECHA_ENTREGA_LOGIS is not null set @estado = 'Entregado'

    set @modificado = 'Modf. Proc. Cancelado'

```

Acción 1

Cuando el estado es 'Pendiente','Picking','Cola_Armado','Armando' cambiamos el estado del pedido y los items a Cancelado. Dentro de los items se le saca la reserva

```

if @estado in ('Pendiente','Picking','Cola_Armado','Armando')
begin
    update PEDI_CABE set ESTADO = 'CAN' , OBS_3 = left(@obs_3,2000), ExternalShippingId =
@plataforma_envio_id+'_CAN',nota = @modificado where TIPO = @tipo and num = @num
    update PEDI_ITEMS set CODI_ESTA = 'REC', RESER = 'N',FECHA_CAMBIO_ESTA = getdate() where TIPO
= @tipo and NUM = @num
end

```

Accion 2

Cuando el pedido se encuentra facturamos modificamos el codigo de envio para evitar que se pueda despachar por el modulo de entrega

```

if @estado = 'Facturado'
begin
    update PEDI_CABE set OBS_3 = left(@obs_3,2000), ExternalShippingId =
@plataforma_envio_id+'_CAN',nota = @modificado where TIPO = @tipo and num = @num
end

```

Acción 3

Cuando el pedido esta entregado agregamos una observación dentro del pedido. Juntamos los datos de la factura y preparamos para emitir una factura al proveedor (pendiente para versión 2)

```
if @estado = 'Entregado'  
begin  
    update PEDI_CABE set OBS_3 = left(@obs_3,2000),nota = @modificado where TIPO = @tipo and num  
= @num  
  
    declare @obs varchar(200) = null  
    select @reci_num =RECI_NUME, @total = total,@remi_num = REMI_NUME from COMP_EMITIDOS where  
TIPO = @TIPO_FC_PICKING and num = @NUM_FC_PICKING  
    select @obs = CONCAT('#IS ',@remi_num,' Factura ',concat(@TIPO_FC_PICKING,'-  
,right(concat('00000000',@NUM_FC_PICKING),8)))  
    end  
end
```

SP - intrasistema_order_detail_get

Con este SP desde Intrasistema consulta el detalle de las facturas para poder subirlo a Mercado Libre

ACTUALIZAR! [@Hernan Reta](#)

•

Código

```
ALTER procedure [dbo].[intrasistema_order_detail_get]
@tipo_fc varchar(4), @num_fc int

as
select r.tipo_fact tipo_fc, r.NUM_FACT
num_fc, trim(replace(replace(replace(replace(ARTICULO,'<i class="fa fa-pencil"',''),CHAR(9),''),CHAR(10),''),CHAR(11),''),CHAR(12),''),CHAR(13),'')) sku,
      trim(replace(replace(replace(replace(replace(a.DESCRIP_ARTI,'<i class="fa fa-pencil"',''),CHAR(9),''),CHAR(10),''),CHAR(11),''),CHAR(12),''),CHAR(13,''))) descripcion
      ,abs(CANT) cantidad
      ,PRECIO precio_unidad,abs(r.PRECIO*CANT)precio_final,r.PORCEN_IVA iva
      from RENG_FAC r
      inner join articulos a on r.ARTICULO=a.COD_ARTICULO
      where r.
```

```
TIPO_FACT = @tipo_fc and r.NUM_FACT = @num_fc
```

Buscamos el detalle de la factura solicitada dentro de la base. Eliminamos los espacios innecesarios del nombre y convertimos caracteres ANSI a nada

```
select r.tipo_fact tipo_fc, r.NUM_FACT
num_fc, trim(replace(replace(replace(replace(ARTICULO,'<i class="fa fa-pencil"',''),CHAR(9),''),CHAR(10),''),CHAR(11),''),CHAR(12),''),CHAR(13),'')) sku,
      trim(replace(replace(replace(replace(replace(a.DESCRIP_ARTI,'<i class="fa fa-pencil"',''),CHAR(9),''),CHAR(10),''),CHAR(11),''),CHAR(12),''),CHAR(13,''))) descripcion
      ,abs(CANT) cantidad
      ,PRECIO precio_unidad,abs(r.PRECIO*CANT)precio_final,r.PORCEN_IVA iva
      from RENG_FAC r
```

```
inner join articulos a on r.ARTICULO=a.COD_ARTICULO  
where r.TIPO_FACT = @tipo_fc and r.NUM_FACT = @num_fc
```

SP - intrasistema_order_post

Con este SP se cargan los pedidos desde intrasistemas a RP

•

Código

```
ALTER procedure [dbo].[intrasistema_order_post]
@plataforma varchar(25)=NULL, -- Por ahora solo ML
@plataforma_orden_id varchar(200) = NULL , --Nro. Operación ML
@plataform
@nickname varchar(30) = null,
@plataforma_publicacion_id varchar (25)= null,
@plataforma_orden_sub_id varchar(200) = null,
@plataforma_envio_id varchar(200) = null,
@plataforma_costo_envio varchar(30) = null,
@plataforma_fecha_orden datetime = null ,
@cod_articulo varchar(30),
@cantidad numeric(12,4),
@precio_sin_iva numeric(12,4),
@precio_con_iva numeric(12,4),
@porcen_descuen numeric(12,4),
@observaciones varchar(200)=NULL,
@nombre_cliente varchar(60)=NULL,
@cate_iva varchar(2)=NULL, --Categoría de IVA ( CF, RI , MO, EX o NC (no clasificado)
@cuit varchar(20)=NULL, --CUIT (Solo para clientes <> CF o NC)
@dni varchar(20)=NULL , --solo para clientes = CF o NC
@domicilio varchar(40)=NULL,
@localidad varchar(30)=NULL,
@provincia varchar(50)=NULL, --- pasar la descripcion de la provincia
@mail varchar(100)=NULL,
@telefono varchar(35)=NULL,
@deposito_id varchar(3) = NULL,
@procesador_negocio_id varchar(200),
@procesador_item_id varchar(25) = null,
@procesador_orden_id varchar(200),
@procesador_item_tipo_id varchar(2)=null,
@procesador_item_erp_id varchar (25) = null,
@procesador_tipo_envio_id varchar(30) = null,
@procesador_proveedor_id varchar(10) = null,
@procesador_fecha_orden datetime,
@procesador_medio_pago varchar(60) = null
as
```

```

set nocount on

declare @codi_provincia varchar(3)
declare @error varchar(800), @mensaje varchar(455) ,
        @obser_2 varchar(200), @obser_3 varchar(2000), @obser_anti varchar(255),
        @CODI_TRANSPOR int, @base varchar(30)

declare @hoy datetime
select @hoy = convert(datetime,convert(varchar,getdate(),112))

if isnull(@deposito_id,'') = ''
begin
    select '1' error,'Debe indicar el codigo de deposito.' descripcion
    return
end

if isnull(@nombre_cliente,'') != '' and isnull(@cate_iva,'') = ''
begin
    select '1' error,'Si se indica el Nombre del cliente, debe indicar la categoria de IVA.'
descripcion
    return
end

if @precio_sin_iva > 500000
begin
    select '1' error , 'El precio unitario sin IVA no puede ser mayor a $500.000' descripcion
    return
end

if isnull(@cate_iva,'') != ''
begin
    if @cate_iva not in ('RI','CF','EX','MO','NC')
    begin
        select '1' error,'El campo categoria de iva solo puede recibir CF, NC, RI , MO o EX .'
descripcion
        return
    end

    if @cate_iva in ('RI','EX','MO') and isnull(@cuit,'') in ('0','')
    begin
        select '1' error,'Si el campo categoria de iva es RI, EX o MO debe informar el CUIT.'
descripcion
        return
    end

    if @cate_iva in ('RI','EX','MO')
    begin
        exec factu_valida_cuit @cuit
        if @@ERROR != 0 return
    end

```

```

if @provincia is not null and @cate_iva in ('RI','EX','MO')
begin
    select @codi_provincia = CODI
    from provincias where NOMBRE = @provincia

    if @codi_provincia is null
    begin
        select '1' error, 'No existe la provincia: ' + @provincia descripcion
        return
    end
end

if @cate_iva in ('RI','EX','MO') and @codi_provincia is null
begin
    select '1' error,'Si el campo categoria de iva es RI, EX o MO debe indicar la
provincia.' descripcion
    return
end

end

if @procesador_orden_id = 0
begin
    select '1' error,'El nro de pedido tecnogo no puede ser cero.' descripcion
    return
end

if isnull(@cod_articulo,'') = ''
begin
    select '1' error,'El codigo de articulo no puede ser vacio.' descripcion
    return
end

--if not exists (select 1 from articulos where cod_articulo = @cod_articulo)
--begin
--    select '1' error,'Codigo de articulo ' + @cod_articulo + ' inexistente.' descripcion
--    return
--end

--if exists (select 1 from articulos a where cod_articulo = @cod_articulo
--           and isnull(activo,'N') = 'N' and left(a.cod_articulo,1) != 'C')
--begin
--    select '1' error, 'Codigo de articulo ' + @cod_articulo + ' esta INACTIVO.' descripcion
--    return
--end

```

```

declare @precio_neto numeric(12,4), @codi_esta varchar(5)

select @precio_neto = isnull(@precio_sin_iva,0) -
   (isnull(@precio_sin_iva,0)*isnull(@porcen_descuen,0)/100)

declare @pedi_num int,
        @cliente int,
        @lista_codi varchar(5),
        @codi_vende varchar(5),
        @renglon int,
        @total_pedido numeric(12,4),
        @porcen_descuen_cabe numeric(12,4),
        @pedido_tipo varchar(3),
        @agrupacion_1 varchar(5),
        @agrupacion_2 varchar(5),
        @agrupacion_3 varchar(5),
        @costo_envio as decimal(12,4),
        @valida as int,
        @renglon_envio as int, @entero as int ,@decimal as decimal(12,4),@fc
varchar(3),@nombre_modificado varchar(60)

IF @deposito_id = 'DPT'
BEGIN
    SET @deposito_id = 'DEP'
END
IF @deposito_id = 'DML'
BEGIN
    SET @deposito_id = 'FUL'
END

if @procesador_item_erp_id = ''
set @procesador_item_erp_id = null

if @plataforma_orden_sub_id = ''
set @plataforma_orden_sub_id = null

if isnull(@plataforma_costo_envio,'') = ''
begin
    set @plataforma_costo_envio = 0
end

set @nombre_modificado = left(@nombre_cliente,50)

select @obser_3 =
TRIM(CONVERT(VARCHAR(2000),'https://api.intrasistema.com/v1/fulfillment/' +TRIM(CONVERT(VARCHAR(10)
,@procesador_negocio_id))+' /prints/' +TRIM(CONVERT(VARCHAR(10),@procesador_orden_id))+' /zpl?
access_token=def502006f031468faf4babd9cd6fa1945f91b1e8b62fe39f9a86e898982522f27b2f66c5f8c8d1d79ca1
7c95cf5d027cb6bcd9657de76584198d7a4cb91ad6434eeef1801cf13901eaa6d0ba8ec84fe7e45ee10479c012069dff69
b3ccca95b1614e5e1505da6c9da46939f1b183af423d4c62a8d2837c1544b7b2a11b0c186e1ca1c6306723c95176a130d37

```

8938e7ae3b0e791b3ab0e356d8805cc27ecba2f961a7725602612d43ffc1844c6fef9791383abc5a94e6de55615689e42
343a046f2a80c06f582986fa1c769c7661aa376276689acfd30210af932ed086a10a5bd32a62af10acfdb ea69815cbdf1f
e96f7136b32549b3244544d8d71355680a3164a61fb484c2dbd1fbab57f5390ee7161bef7a753144f4f5ada8ac2488849e
fe6984a1828cd6af7b361227aa49fcf5efeee05e070f72461162c'))

```
SELECT @base = base
FROM DBO.INTRASISTEMA_LOC
WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

if @base = 'Central'
begin
    if isnull(@nombre_cliente,'') = ''
        SELECT @pedido_tipo=tipo,@cliente=cliente,@lista_codi=lista_codi
        FROM DBO.INTRASISTEMA_LOC
        WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

    else
        begin
            if @cate_iva in ('RI','EX','MO')
                begin
                    select @cliente = num_cliente
                    from clientes where replace(cuit,'-','') = replace(@cuit,'-','')

                    SELECT @pedido_tipo=tipo,@lista_codi=lista_codi
                    FROM DBO.INTRASISTEMA_LOC
                    WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

                    if isnull(@cliente,0) = 0
                        begin

                            SELECT @pedido_tipo=tipo,@lista_codi=lista_codi, @agrupacion_2 =
                            agr_2,@agrupacion_3 = agr_3,@fc = fc_a_default
                            FROM DBO.INTRASISTEMA_LOC
                            WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and
                            @plataforma=plataforma

                            select @cliente = max(num_cliente) from clientes
                            select @cliente = isnull(@cliente,0)+1

                            insert into clientes
                            (num_cliente, razon, cuit, iva, estado, domicilio, localidad,
                            provincia, mail, telefono, fecha_alta, user_alta, codi_vende, PORCEN_DESCUEN, agru_1, agru_2, agru_3,
                            NOM_FANTASIA, COD_COMP )

                            select isnull(@cliente,0),@nombre_modificado,@cuit,
                                case when @cate_iva = 'RI' then '1'
                                    when @cate_iva = 'EX' then '3'
                                    when @cate_iva = 'MO' then '5' else '4' end IVA,
```

```

        'A',@domicilio,@localidad,@codi_provincia,
        @mail,@telefono,getdate(),substring(suser_sname(),1,20),
        @codi_vende codi_vende, 0, @agrupacion_1,@agrupacion_2,@agrupacion_3,
        @nombre_modificado,@fc
    if @@error != 0 return
end

end
else
begin

SELECT @pedido_tipo=tipo,@cliente=cliente,@lista_codi=lista_codi
FROM DBO.INTRASISTEMA_LOC
WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

end
end

select @pedi_num = num
from pedi_cabe
where REFERENCIA = convert(varchar,@procesador_orden_id)

if isnull(@pedi_num,0) > 0
begin
    if exists (select 1 from pedi_items
               where tipo = @pedido_tipo and num = @pedi_num and articulo = @cod_articulo and
OBS2 = concat('Articulo Org: ',isnull(@procesador_item_erp_id,@procesador_item_id))) --PEDIDO DE
MALARIA
        begin
            select '0' error ,concat(@pedido_tipo , '-' ,@pedi_num) nro_pedido
            return
        end
    end
end

select @lista_codi = lista_codi ,
       @codi_vende = codi_vende
from clientes where num_cliente = @cliente

begin transaction

if isnull(@pedi_num,0) = 0
begin

select @pedi_num = prox_num from comp where cod_comp = @pedido_tipo--'DTE' ---CAMBIADO

```

PARA FULL

```
update comp set prox_num = prox_num + 1 where cod_comp = @pedido_tipo --'DTE' ----CAMBIO  
PARA FFULL
```

```
insert into pedi_cabe (tipo,num,fecha,cliente,lista_codi,obs,mone,mone_coti,  
user_alta,estado,referencia,condi_venta,total,codi_vende,  
porcen_descuen,lugar_entre,fecha_entre,OBS_2,OBS_3, fecha_alta,  
TELEFONO_CLI , MAIL_CLI,CODI_TRANSPOR,CODI_LEGAJO,  
CODI_LEGAJ02,OC_NUME,PERCEP_IB_PROVIN,URL_ML,
```

NOMBRE_CLI,DIRECCION_CLI,LOCALIDAD_CLI,PROVINCIA_CLI,TIPO_DOC_CLI,NUM_DOC_CLI,ReceiverAddress,ExternalShippingId

,BLOQUEAR,CANT_MAX_ITEMS_WEB, ID_PEDIDO_ALEPH, NICKNAME_CLI, PackId, LogisticType, LogisticName)

```
select  
@pedido_tipo,@pedi_num,convert(varchar,isnull(@procesador_fecha_orden,@plataforma_fecha_orden),23)  
,@cliente,isnull(@lista_codi,'MELI') codi_lista, --CAMBIADO PARA FULL  
'Nickname:  
'+trim(isnull(@plataforma_nickname,'')),'PES',1,substring(suser_sname(),1,20),'PEN',  
substring(convert(varchar,@procesador_orden_id),1,50),  
NULL condิ_venta,0, null,--@codi_vende  
0 porcen_descuen, NULL codi_lugar, NULL fecha_entre,  
substring('Cliente: '+isnull(@nombre_modificado,'')+'' Direccion entrega:  
'+isnull(@domicilio,'')+'' Telefono entrega: '+isnull(@telefono,'')+'' Contacto entrega:  
'+isnull(@nombre_modificado,'')+'' Nombre empresa: DNI '+isnull(@dni,'')+'' Pack ID:  
'+cast(isnull(isnull(@plataforma_orden_sub_id,@plataforma_envio_id),'') as varchar)+'' Notas:  
,1,200),null ,getdate(), @telefono, @mail,@CODI_TRANSPOR,  
'SS1','SP1',isnull(@plataforma_orden_sub_id,@plataforma_orden_id),@cate_iva,@obser_3,@nombre_modificado,@domicilio,@localidad,@provincia,'DNI',isnull(@dni,@cuit),@domicilio,@plataforma_envio_id  
, 'S','1',@procesador_orden_id,@plataforma_nickname,@plataforma_orden_sub_id,case when  
@procesador_tipo_envio_id = '1' then '3'  
when @procesador_tipo_envio_id = '6' then '5'  
when @procesador_tipo_envio_id = '5' then '6'  
else '1' end  
,case when @procesador_tipo_envio_id = '1' then 'Mercado Envios 2'  
when @procesador_tipo_envio_id = '4' then 'Flex'  
when @procesador_tipo_envio_id = '6' then 'FULL'  
else 'Mercado Envios 2' end  
if @@error<>0  
begin  
rollback transaction  
return  
end  
end  
  
declare @max_porcen_iva numeric(12,4)
```

```

exec facti_iva_refer_consul_2 @max_porcen_iva out

select @renglon = max(renglon) from pedi_items
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL


insert into pedi_items (tipo,num,renglon,articulo,cant,precio,
cant_pend,codi_esta,
bultos,porcen_iva,porcen_descuen,tipo_bonif,
RESER,DEPO_RESER,COSTO_UNI,
OBS1,OBS2,OBS4,DEPO_RETIRO,id_pedido_aleph,ExternalOrderId,
PackId,id_item_aleph,SellerAccount,CANTI_ORIGINAL)
select @pedido_tipo tipo,@pedi_num,isnull(@renglon,0)+1,
@cod_articulo,
@cantidad,
@precio_con_iva,
@cantidad,'PRE',
0,--bultos
case when a.porcen_iva = 0 and a.exento = 'N' then @max_porcen_iva
else a.porcen_iva end,
@porcen_descuen,'D',
'S' reser, @deposito_id, a.precio_uni
,'IntegrationCode:' +trim(isnull(@plataforma_publicacion_id,'')), 'Articulo Org:
'+isnull(@procesador_item_erp_id,@procesador_item_id) , 'Cantidad Original: ' +cast(@cantidad as
varchar),
@deposito_id,@procesador_orden_id,@plataforma_orden_id,isnull(@plataforma_orden_sub_id,@plataforma
_orden_id),
@procesador_item_id,@procesador_negocio_id,@cantidad
from articulos a where a.cod_articulo = @cod_articulo
if @@error<>0
begin
    rollback transaction
    return
end

if cast(@plataforma_costo_envio as numeric(12,4)) > 0

begin
if @plataforma_costo_envio like '%%'
begin
select @entero = convert(int ,LEFT(@plataforma_costo_envio, CHARINDEX('.',@plataforma_costo_envio) - 1 )
```

```

select @decimal = convert(decimal(12,4),concat('0.',SUBSTRING(@plataforma_costo_envio,
(CHARINDEX('. ',@plataforma_costo_envio)+1),4)))

set @costo_envio = convert(numeric(12,4),@entero + @decimal)
end
else
set @costo_envio = @plataforma_costo_envio

--set @cod_articulo = 'A40'
select @valida = 1 from PEDI_ITEMS where tipo = @pedido_tipo and num = @pedi_num and ARTICULO
= 'A40' and OBS3 = @plataforma_envio_id
if isnull(@valida,0) <> 1

begin

    select @renglon_envio = max(renglon) from PEDI_ITEMS where tipo = @pedido_tipo and num =
@pedi_num

    insert into pedi_items (tipo,num,renglon,articulo,cant,precio,
    cant_pend,codi_esta,
    bultos,porcen_iva,porcen_descuen,tipo_bonif,
    RESER,DEPO_RESER,COSTO_UNI,
    OBS1,OBS2,OBS4,DEPO_RETIRO,id_pedido_aleph,ExternalOrderId,
    PackId,id_item_aleph,SellerAccount,CANTI_ORIGINAL,OBS3)
    select @pedido_tipo tipo,@pedi_num,isnull(@renglon_envio,0)+1,
    'A40',
    1,
    @costo_envio ,
    1,isnull(@codi_esta,'PRE'),
    0,--bultos
    case when a.porcen_iva = 0 and a.exento = 'N' then @max_porcen_iva
        else a.porcen_iva end,
    @porcen_descuen,'D',
    'S' reser, @deposito_id, a.precio_uni
    , 'IntegrationCode:'+trim(isnull(@plataforma_publicacion_id,'')), 'Articulo Org:
'+isnull(@procesador_item_erp_id,@procesador_item_id) , 'Cantidad Original: '+cast(@cantidad as
varchar),
    @deposito_id,@procesador_orden_id,@plataforma_orden_id,isnull(@plataforma_orden_sub_id,@plataforma
_orden_id),
    @procesador_item_id,@procesador_negocio_id,@cantidad,@plataforma_envio_id
    from articulos a where a.cod_articulo = @cod_articulo
    if @@error<>0
    begin
        rollback transaction
        return
    end
    end
    end

```

```

select @total_pedido = sum(cant* (precio-(precio*porcen_descuen/100)))
from pedi_items
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL

select @porcen_descuen_cabe = ISNULL(porcen_descuen,0)
from PEDI_CABE where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL

if @porcen_descuen_cabe > 0
    select @total_pedido = @total_pedido - (@total_pedido * @porcen_descuen_cabe / 100)

update pedi_cabe
set total = round(@total_pedido,2)
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL
if @@error<>0
begin
    rollback transaction
    return
end

if @mensaje is not null
begin
    select @mensaje = @mensaje + 'Pedido Tecnogo ' + convert(varchar,@pedi_num)
    exec factu_mensajes_usuarios_alta 'timer_tecnogo','dba_luciano',@mensaje
end

commit transaction
end

if @base = 'Locales'
begin

SELECT @pedido_tipo=tipo,@cliente=cliente,@lista_codi=lista_codi
      FROM DBO.INTRASISTEMA_LOC
      WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

select @pedi_num = num
      from factu_full_locales.dbo.pedi_cabe
      where REFERENCIA = convert(varchar,@procesador_orden_id)

if isnull(@pedi_num,0) > 0
begin
    if exists (select 1 from factu_full_locales.dbo.pedi_items
               where tipo = @pedido_tipo and num = @pedi_num and articulo = @cod_articulo and
OBS2 = concat('Articulo Org: ',isnull(@procesador_item_erp_id,@procesador_item_id))) --PEDIDO DE
MALARIA
        begin
            select '0' error ,concat(@pedido_tipo , '-' ,@pedi_num) nro_pedido

```

```

        return
    end
end

select @lista_codi = lista_codi ,
       @codi_vende = codi_vende
from factu_full_locales.dbo.clientes where num_cliente = @cliente

begin transaction

if isnull(@pedi_num,0) = 0
begin

    select @pedi_num = prox_num from factu_full_locales.dbo.comp where cod_comp =
@pedido_tipo--'DTE' ---CAMBIADO PARA FULL
    update factu_full_locales.dbo.comp set prox_num = prox_num + 1 where cod_comp =
@pedido_tipo --'DTE' ----CAMBIO PARA FFULL

    insert into factu_full_locales.dbo.pedi_cabe
(tipo,num,fecha,cliente,lista_codi,obs,mone,mone_coti,
user_alta,estado,referencia,condi_venta,total,codi_vende,
porcen_descuen,lugar_entre,fecha_entre,OBS_2,OBS_3, fecha_alta,
TELEFONO_CLI , MAIL_CLI,CODI_TRANSPO,CODI_LEGATO,
CODI_LEGATO2,OC_NUME,PERCEP_IB_PROVIN,
NOMBRE_CLI,DIRECCION_CLI,LOCALIDAD_CLI,PROVINCIA_CLI,TIPO_DOC_CLI,NUM_DOC_CLI,ReceiverAddress,ExternalShippingId
,BLOQUEAR,CANT_MAX_ITEMS_WEB,ID_PEDIDO_ALEPH,NICKNAME_CLI,PackId)

    select
@pedido_tipo,@pedi_num,convert(varchar,isnull(@procesador_fecha_orden,@plataforma_fecha_orden),23),
@cliente,isnull(@lista_codi,'MELI') codi_lista, --CAMBIADO PARA FULL
'Nickname:
'+trim(isnull(@plataforma_nickname,'')),'PES',1,substring(suser_sname(),1,20),'PEN',
substring(convert(varchar,@procesador_orden_id),1,50),
NULL condิ_venta,0, null,--@codi_vende
0 porcen_descuen, NULL codi_lugar, NULL fecha_entre,
substring('Cliente: '+isnull(@nombre_modificado,'')+Direccion entrega:
'+isnull(@domicilio,'')+ Telefono entrega: '+isnull(@telefono,'')+ Contacto entrega:
'+isnull(@nombre_modificado,'')+ Nombre empresa: DNI '+isnull(@dni,'')+ Pack ID:
'+cast(isnull(isnull(@plataforma_orden_sub_id,@plataforma_envio_id),'') as varchar)+ Notas:
',1,200),null ,getdate(), @telefono, @mail,@CODI_TRANSPO,
'SS1','SP1',isnull(@plataforma_orden_sub_id,@plataforma_orden_id),@cate_iva,@nombre_modificado,@domicilio,@localidad,@provincia,'DNI',isnull(@dni,@cuit),@domicilio,@plataforma_envio_id
,'S','1',@procesador_orden_id,@plataforma_nickname,@plataforma_orden_sub_id
if @@error<>0
begin

```

```

        rollback transaction
        return
    end

end

--declare @max_porcen_iva numeric(12,4)
exec factu_full_locales.dbo.facti_iva_refer_consul_2 @max_porcen_iva out


select @renglon = max(renglon) from factu_full_locales.dbo.pedi_items
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL


insert into factu_full_locales.dbo.pedi_items (tipo,num,renglon,articulo,cant,precio,
cant_pend,codi_esta,
bultos,porcen_iva,porcen_descuen,tipo_bonif,
RESER,DEPO_RESER,COSTO_UNI,
OBS1,OBS2,OBS4,DEPO_RETIRO,id_pedido_aleph,ExternalOrderId,
PackId,id_item_aleph,SellerAccount)
select @pedido_tipo tipo,@pedi_num, isnull(@renglon,0)+1,
@cod_articulo,
@cantidad,
@precio_con_iva,
@cantidad,'PEN',
0,--bultos
case when a.porcen_iva = 0 and a.exento = 'N' then @max_porcen_iva
else a.porcen_iva end,
@porcen_descuen,'D',
'S' reser, @deposito_id, a.precio_uni
,'IntegrationCode:' +trim(isnull(@plataforma_publicacion_id,'')), 'Articulo Org:
'+isnull(@procesador_item_erp_id,@procesador_item_id) , 'Cantidad Original: '+cast(@cantidad as
varchar),
@deposito_id,@procesador_orden_id,@plataforma_orden_id,isnull(@plataforma_orden_sub_id,@plataforma
_orden_id),
@procesador_item_id,@procesador_negocio_id
from factu_full_locales.dbo.articulos a where a.cod_articulo = @cod_articulo
if @@error<>0
begin
    rollback transaction
    return
end

select @total_pedido = sum(cant* (precio-(precio*porcen_descuen/100)))
from factu_full_locales.dbo.pedi_items
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL

```

```

select @porcen_descuen_cabe = ISNULL(porcen_descuen,0)
from factu_full_locales.dbo.PEDI_CABE where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO
PARA FULL

if @porcen_descuen_cabe > 0
    select @total_pedido = @total_pedido - (@total_pedido * @porcen_descuen_cabe / 100)

update factu_full_locales.dbo.pedi_cabe
set total = round(@total_pedido,2)
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL
if @@error<>0
begin
    rollback transaction
    return
end

commit transaction
end

select '0' error ,concat(@pedido_tipo , '-' ,@pedi_num) nro_pedido

```

Definición de campos

```

@plataforma varchar(25)=NULL, -- Por ahora solo ML
@plataforma_orden_id varchar(200) = NULL , --Nro. Operación ML
@plataforma_nickname varchar(30) = null,
@plataforma_publicacion_id varchar (25)= null,
@plataforma_orden_sub_id varchar(200) = null,
@plataforma_envio_id varchar(200) = null,
@plataforma_costo_envio varchar(30) = null,
@plataforma_fecha_orden datetime = null ,
@cod_articulo varchar(30),
@cantidad numeric(12,4),
@precio_sin_iva numeric(12,4),
@precio_con_iva numeric(12,4),
@porcen_descuen numeric(12,4),
@observaciones varchar(200)=NULL,
@nombre_cliente varchar(60)=NULL,
@cate_iva varchar(2)=NULL, --Categoría de IVA ( CF, RI , MO, EX o NC (no clasificado)
@cuit varchar(20)=NULL, --CUIT (Solo para clientes <> CF o NC)
@dni varchar(20)=NULL , --solo para clientes = CF o NC
@domicilio varchar(40)=NULL,
@localidad varchar(30)=NULL,

```

```

@provincia varchar(50)=NULL, --- pasar la descripcion de la provincia
@mail varchar(100)=NULL,
@telefono varchar(35)=NULL,
@deposito_id varchar(3) = NULL,
@procesador_negocio_id varchar(200),
@procesador_item_id varchar(25) = null,
@procesador_orden_id varchar(200),
@procesador_item_tipo_id varchar(2)=null,
@procesador_item_erp_id varchar (25) = null,
@procesador_tipo_envio_id varchar(30) = null,
@procesador_proveedor_id varchar(10) = null,
@procesador_fecha_orden datetime,
@procesador_medio_pago varchar(60) = null

```

Declaracion de primeras variables

```

declare @codi_provincia varchar(3)
declare @error varchar(800), @mensaje varchar(455) ,
        @obser_2 varchar(200), @obser_3 varchar(2000), @obser_anti varchar(255),
        @CODI_TRANSPOR int, @base varchar(30)

declare @hoy datetime
select @hoy = convert(datetime,convert(varchar,getdate()),112)

```

Filtro de errores

El codigo de deposito no puede ser vacio

```

if isnull(@deposito_id,'') = ''
begin
    select '1' error,'Debe indicar el codigo de deposito.' descripcion
    return
end

```

Si se envia el numero de cliente tiene que enviar el tipo de iva

```

if isnull(@nombre_cliente,'') != '' and isnull(@cate_iva,'') = ''
begin
    select '1' error,'Si se indica el Nombre del cliente, debe indicar la categoria de IVA.'
    descripcion
    return
end

```

Se agrego este filtro por un error que tenia intrasistema

```

if @precio_sin_iva > 500000
begin
    select '1' error , 'El precio unitario sin IVA no puede ser mayor a $500.000' descripcion

```

```
    return  
end
```

El campo de categoría de iva tiene que ser uno de los definidos en el RP

```
if isnull(@cate_iva,'') != ''  
begin  
    if @cate_iva not in ('RI','CF','EX','MO','NC')  
    begin  
        select '1' error,'El campo categoria de iva solo puede recibir CF, NC, RI , MO o EX .'  
descripcion  
    return  
end
```

Si la categoria de iva es responsable inscripto, monotributo o extento tiene que enviar el cuit.
Tambien la provincia tiene que enviarse y el nombre debe ser el mismo que en RP

```
if @cate_iva in ('RI','EX','MO') and isnull(@cuit,'') in ('0','')  
begin  
    select '1' error,'Si el campo categoria de iva es RI, EX o MO debe informar el CUIT.'  
descripcion  
return  
end  
  
if @cate_iva in ('RI','EX','MO')  
begin  
exec factu_valida_cuit @cuit  
if @@ERROR != 0 return  
end  
  
if @provincia is not null and @cate_iva in ('RI','EX','MO')  
begin  
select @codi_provincia = CODI  
from provincias where NOMBRE = @provincia  
  
if @codi_provincia is null  
begin  
    select '1' error, 'No existe la provincia: ' + @provincia descripcion  
    return  
end  
end  
  
if @cate_iva in ('RI','EX','MO') and @codi_provincia is null  
begin  
    select '1' error,'Si el campo categoria de iva es RI, EX o MO debe indicar la provincia.'  
descripcion  
return  
end  
end
```

El numero de intrasistema no puede ser 0

```
if @procesador_orden_id = 0
begin
    select '1' error,'El nro de pedido tecnogo no puede ser cero.' descripcion
    return
end
```

Siempre se tiene que enviar el codigo de producto

```
if isnull(@cod_articulo,'') = ''
begin
    select '1' error,'El codigo de artoculo no puede ser vacio.' descripcion
    return
end
```

Segunda definicion de variables

```
declare @precio_neto numeric(12,4), @codi_esta varchar(5)

select @precio_neto = isnull(@precio_sin_iva,0) -
                    (isnull(@precio_sin_iva,0)*isnull(@porcen_descuen,0)/100)

declare @pedi_num int,
        @cliente int,
        @lista_codi varchar(5),
        @codi_vende varchar(5),
        @renglon int,
        @total_pedido numeric(12,4),
        @porcen_descuen_cabe numeric(12,4),
        @pedido_tipo varchar(3),
        @agrupacion_1 varchar(5),
        @agrupacion_2 varchar(5),
        @agrupacion_3 varchar(5),
        @costo_envio as decimal(12,4),
        @valida as int,
        @renglon_envio as int, @entero as int ,@decimal as decimal(12,4),@fc
varchar(3),@nombre_modificado varchar(60)
```

Modificaciones

Se cambia los codigos de deposito, quedo por un error en el envio de Intrasistema

```
IF @deposito_id = 'DPT'
BEGIN
    SET @deposito_id = 'DEP'
END
IF @deposito_id = 'DML'
```

```

BEGIN
    SET @deposito_id = 'FUL'
END

```

Cambiamos valores enviados en vacio a null para no interferir con el código. La variable nombre modificado se creo para acortar los caracteres de los nombres del cliente (la columna de nombres tiene un máximo)

```

if @procesador_item_erp_id = ''
    set @procesador_item_erp_id = null

if @plataforma_orden_sub_id = ''
    set @plataforma_orden_sub_id = null

if isnull(@plataforma_costo_envio,'') = ''
begin
    set @plataforma_costo_envio = 0
end

set @nombre_modificado = left(@nombre_cliente,50)

```

Link de etiqueta

Creamos el link de la etiqueta de Intrasistema remplazando el numero de negocio y numero de pedido en intra

```

select @obser_3 =
TRIM(CONVERT(VARCHAR(2000),'https://api.intrasistema.com/v1/fulfillment/' +TRIM(CONVERT(VARCHAR(10),@procesador_negocio_id))+'/prints/' +TRIM(CONVERT(VARCHAR(10),@procesador_orden_id))+'/zpl?
access_token=def502006f031468faf4babd9cd6fa1945f91b1e8b62fe39f9a86e898982522f27b2f66c5f8c8d1d79ca17c95cf5
d027cb6bcd9657de76584198d7a4cb91ad6434eef1801cf13901eaa6d0ba8ec84fe7e45ee10479c012069dff69b3cca95b1614e5
e1505da6c9da46939f1b183af423d4c62a8d2837c1544b7b2a11b0c186e1ca1c6306723c95176a130d378938e7ae3b0e791b3ab0e
356d8805cc27ecba2f961a7725602612d43ffc1844c6dfef9791383abc5a94e6de55615689e42343a046f2a80c06f582986fa1c76
9c7661aa376276689acfd30210af932ed086a10a5bd32a62af10acfdbbea69815cbdf1fe96f7136b32549b3244544d8d71355680a3
164a61fb484c2dbd1fbab57f5390ee7161bef7a753144f4f5ada8ac2488849efe6984a1828cd6af7b361227aa49fc5efeee05e07
0f72461162c' ))

```

Cambio de base

Buscamos la tabla Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-15834>) que tipo de base tiene que usarse para el tipo de pedido. Ej: los pedidos de mercado libre se dan de alta en Central pero los pedidos de Pedidos Ya se dan de alta en Locales

```

SELECT @base = base
    FROM DBO.INTRASISTEMA_LOC
    WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

```

Pedido Central

Si el pedido hay que crearlo en la base de central primero validamos el nombre enviado, si el nombre esta vacio usamos los valores de deposito, bussines id y plataforma para buscar en la

tabla de Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-15834>) que tipo de pedido hay que crear.

```
if @base = 'Central'
begin
    if isnull(@nombre_cliente, '') = ''
        SELECT @pedido_tipo=tipo,@cliente=cliente,@lista_codi=list_a_codi
        FROM DBO.INTRASISTEMA_LOC
        WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

    else
```

Si el cliente es reposable inscripto, extento o monotributista buscamos el cuit dentro de la base. Si no esta dado de alta procedemos a crearlo

```
begin

    if @cate_iva in ('RI','EX','MO')
    begin
        select @cliente = num_cliente
        from clientes where replace(cuit,'-','') = replace(@cuit,'-','')

        SELECT @pedido_tipo=tipo,@lista_codi=list_a_codi
        FROM DBO.INTRASISTEMA_LOC
        WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

        if isnull(@cliente,0) = 0
        begin

            SELECT @pedido_tipo=tipo,@lista_codi=list_a_codi, @agrupacion_2 = agr_2,@agrupacion_3 =
            agr_3,@fc = fc_a_default
            FROM DBO.INTRASISTEMA_LOC
            WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

            select @cliente = max(num_cliente) from clientes
            select @cliente = isnull(@cliente,0)+1

            insert into clientes
            (num_cliente, razon, cuit, iva, estado, domicilio, localidad,
            provincia, mail, telefono, fecha_alta, user_alta, codi_vende, PORCEN_DESCUEN, agru_1, agru_2, agru_3,
            NOM_FANTASIA, COD_COMP )

            select isnull(@cliente,0),@nombre_modificado,@cuit,
            case when @cate_iva = 'RI' then '1'
                when @cate_iva = 'EX' then '3'
                when @cate_iva = 'MO' then '5' else '4' end IVA,
            'A',@domicilio,@localidad,@codi_provincia,
            @mail,@telefono,getdate(),substring(suser_sname(),1,20),
            @codi_vende codi_vende, 0, @agrupacion_1,@agrupacion_2,@agrupacion_3,
```

```

@nombre_modificado,@fc
if @@error != 0 return
end

end

```

Si el cliente no cumple ninguna de las condiciones anteriores seleccionamos el cliente definido en la tabla Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-15834>)

```

SELECT @pedido_tipo=tipo,@cliente=cliente,@lista_codi=listas_codi
FROM DBO.INTRASISTEMA_LOC
WHERE @deposito_id=depo and @procesador_negocio_id=id_negocio and @plataforma=plataforma

```

Alta pedido

Buscamos el numero de id de intrasistema dentro de la base

```

select @pedi_num = num
from pedi_cabe
where REFERENCIA = convert(varchar,@procesador_orden_id)

```

Si encontramos un pedido con el numero de intrasistema en la base validamos que no sea el mismo articulo. Asi evitamos crear un pedido con articulos repetidos

```

if isnull(@pedi_num,0) > 0
begin
    if exists (select 1 from pedi_items
                where tipo = @pedido_tipo and num = @pedi_num and articulo = @cod_articulo and OBS2 =
concat('Articulo Org: ',isnull(@procesador_item_erp_id,@procesador_item_id))) --PEDIDO DE MALARIA
        begin
            select '0' error ,concat(@pedido_tipo ,'-',@pedi_num) nro_pedido
            return
        end
    end
end

```

Seleccionamos el vendedor y el codigo de lista para el cliente

```

select @lista_codi = lista_codi ,
       @codi_vende = codi_vende
  from clientes where num_cliente = @cliente

```

Damos de alta el pedido, en caso que el pedido ya este cargado y hay que agregar articulos se saltea este paso y sigue en el alta de articulos

```

begin transaction

if isnull(@pedi_num,0) = 0
begin

```

```

select @pedi_num = prox_num from comp where cod_comp = @pedido_tipo
update comp set prox_num = prox_num + 1 where cod_comp = @pedido_tipo

insert into pedi_cabe (tipo,num,fecha,cliente,lista_codi,obs,mone,mone_coti,
user_alta,estado,referencia,condi_venta,total,codi_vende,
porcen_descuen,lugar_entre,fecha_entre,OBS_2,OBS_3, fecha_alta,
TELEFONO_CLI , MAIL_CLI,CODI_TRANSPO,CODI_LEGATO,
CODI_LEGATO2,OC_NUME,PERCEP_IB_PROVIN,URL_ML,
NOMBRE_CLI,DIRECCION_CLI,LOCALIDAD_CLI,PROVINCIA_CLI,TIPO_DOC_CLI,NUM_DOC_CLI,ReceiverAddress,ExternalShippingId
,BLOQUEAR,CANT_MAX_ITEMS_WEB,ID_PEDIDO_ALEPH,NICKNAME_CLI,PackId,LogisticType,LogisticName)

```

```

select
@pedido_tipo,@pedi_num,convert(varchar,isnull(@procesador_fecha_orden,@plataforma_fecha_orden),23),@cliente,isnull(@lista_codi,'MELI') codi_lista, --CAMBIADO PARA FULL
'Nickname:
'+trim(isnull(@plataforma_nickname,'')),'PES',1,substring(suser_sname(),1,20),'PEN',
substring(convert(varchar,@procesador_orden_id),1,50),
NULL condิ_venta,0, null,--@codi_vende
0 porcen_descuen, NULL codi_lugar, NULL fecha_entre,
substring('Cliente: '+isnull(@nombre_modificado,'')+'' Direccion entrega:
'+isnull(@domicilio,'')+'' Telefono entrega: '+isnull(@telefono,'')+'' Contacto entrega:
'+isnull(@nombre_modificado,'')+'' Nombre empresa: DNI '+isnull(@dni,'')+'' Pack ID:
'+cast(isnull(isnull(@plataforma_orden_sub_id,@plataforma_envio_id),'') as varchar)+'' Notas:
',1,200),null ,getdate(), @telefono, @mail,@CODI_TRANSPO,
'SS1','SP1',isnull(@plataforma_orden_sub_id,@plataforma_orden_id),@cate_iva,@obser_3,@nombre_modificado,@domicilio,@localidad,@provincia,'DNI',isnull(@dni,@cuit),@domicilio,@plataforma_envio_id
,'S','1',@procesador_orden_id,@plataforma_nickname,@plataforma_orden_sub_id,case when
@procesador_tipo_envio_id = '1' then '3'
when @procesador_tipo_envio_id = '6' then '5'
when @procesador_tipo_envio_id = '5' then '6'
else '1' end
,case when @procesador_tipo_envio_id = '1' then 'Mercado Envios 2'
when @procesador_tipo_envio_id = '4' then 'Flex'
when @procesador_tipo_envio_id = '6' then 'FULL'
else 'Mercado Envios 2' end
if @@error<>0
begin
    rollback transaction
    return
end

```

Consultamos el iva que configurado en la base

```

declare @max_porcen_iva numeric(12,4)
exec facti_iva_refer_consul_2 @max_porcen_iva out

```

Buscamos el numero de renglón mas grande para asociar a un nuevo articulo

```
select @renglon = max(renglon) from pedi_items  
where tipo = @pedido_tipo and num = @pedi_num
```

Cargamos el articulo dentro del pedido

```
insert into pedi_items (tipo,num,renglon,articulo,cant,precio,  
cant_pend,codi_esta,  
bultos,porcen_iva,porcen_descuen,tipo_bonif,  
RESER,DEPO_RESER,COSTO_UNI,  
OBS1,OBS2,OBS4,DEPO_RETIRO,id_pedido_aleph,ExternalOrderId,  
PackId,id_item_aleph,SellerAccount,CANTI_ORIGINAL)  
select @pedido_tipo tipo,@pedi_num, isnull(@renglon,0)+1,  
@cod_articulo,  
@cantidad,  
@precio_con_iva,  
@cantidad,'PRE',  
0,--bultos  
case when a.porcen_iva = 0 and a.exento = 'N' then @max_porcen_iva  
else a.porcen_iva end,  
@porcen_descuen,'D',  
'S' reser, @deposito_id, a.precio_uni  
, 'IntegrationCode:'+trim(isnull(@plataforma_publicacion_id,'')), 'Articulo Org:  
'+isnull(@procesador_item_erp_id,@procesador_item_id) , 'Cantidad Original: '+cast(@cantidad as varchar),  
  
@deposito_id,@procesador_orden_id,@plataforma_orden_id,isnull(@plataforma_orden_sub_id,@plataforma_orden_id),  
@procesador_item_id,@procesador_negocio_id,@cantidad  
from articulos a where a.cod_articulo = @cod_articulo  
if @@error<>0  
begin  
    rollback transaction  
    return  
end
```

Costo de envío

Si el pedido tiene un costo de envío se cargar como un articulo nuevo en el pedido, buscamos si el costo de envío es mayor a 0 y si se envía en modo texto lo pasamos a numero

```
if cast(@plataforma_costo_envio as numeric(12,4)) > 0  
  
begin  
    if @plataforma_costo_envio like '%.%'  
begin  
        select @entero = convert(int ,LEFT(@plataforma_costo_envio, CHARINDEX('.', @plataforma_costo_envio) - 1))  
        )  
        select @decimal = convert(decimal(12,4),concat('0.', SUBSTRING(@plataforma_costo_envio,
```

```
(CHARINDEX('.',@plataforma_costo_envio)+1),4)))
```

```
set @costo_envio = convert(numeric(12,4),@entero + @decimal)
end
else
set @costo_envio = @plataforma_costo_envio
```

Validamos que el pedido no tenga un costo de envío anterior, si no tiene uno creado se agrega al pedido

```
select @valida = 1 from PEDI_ITEMS where tipo = @pedido_tipo and num = @pedi_num and ARTICULO = 'A40'
and OBS3 = @plataforma_envio_id

if isnull(@valida,0) <> 1

begin

select @renglon_envio = max(renglon) from PEDI_ITEMS where tipo = @pedido_tipo and num = @pedi_num

insert into pedi_items (tipo,num,renglon,articulo,cant,precio,
cant_pend,codi_esta,
bultos,porcen_iva,porcen_descuen,tipo_bonif,
RESER,DEPO_RESER,COSTO_UNI,
OBS1,OBS2,OBS4,DEPO_RETIRO,id_pedido_aleph,ExternalOrderId,
PackId,id_item_aleph,SellerAccount,CANTI_ORIGINAL,OBS3)
select @pedido_tipo tipo,@pedi_num,isnull(@renglon_envio,0)+1,
'A40',
1,
@costo_envio ,
1,isnull(@codi_esta,'PRE'),
0,--bultos
case when a.porcen_iva = 0 and a.exento = 'N' then @max_porcen_iva
else a.porcen_iva end,
@porcen_descuen,'D',
'S' reser, @deposito_id, a.precio_uni
,'IntegrationCode:'+trim(isnull(@plataforma_publicacion_id,'')), 'Articulo Org:
'+isnull(@procesador_item_erp_id,@procesador_item_id) , 'Cantidad Original: '+cast(@cantidad as varchar),
@deposito_id,@procesador_orden_id,@plataforma_orden_id,isnull(@plataforma_orden_sub_id,@plataforma_orden_id),
@procesador_item_id,@procesador_negocio_id,@cantidad,@plataforma_envio_id
from articulos a where a.cod_articulo = @cod_articulo
if @@error<>0
begin
rollback transaction
return
end
end
end
```

Total del pedido

Se calcula el total del pedido sumando el precio final de los artículos y restando si tiene descuento aplicado

```
select @total_pedido = sum(cant* (precio-(precio*porcen_descuen/100)))
       from pedi_items
      where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL

select @porcen_descuen_cabe = ISNULL(porcen_descuen,0)
       from PEDI_CABE where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL

if @porcen_descuen_cabe > 0
    select @total_pedido = @total_pedido - (@total_pedido * @porcen_descuen_cabe / 100)

update pedi_cabe
set total = round(@total_pedido,2)
where tipo = @pedido_tipo and num = @pedi_num --CAMBIADO PARA FULL
if @@error<>0
begin
    rollback transaction
    return
end
```

SP - intrasistema_order_status

Con este SP desde Intrasistema consultan el estado de los pedidos desde Intrasistema

•

Código

```
ALTER procedure [dbo].[intrasistema_order_status]
AS
set nocount on
```

Traemos la informacion de pedidos de los últimos 20 dias de los pedidos que tengan un numero de referencia asociado

```
select i.id_negocio, p.tipo, num, FECHA_ENTREGA_LOGIS , USUARIO_ENTREGA , FECHA , ID_PEDIDO_ALEPH ,
LOTE_FECHA , LOTE_USUARIO, USUARIO_ARMADO, FECHA_ARMADO
from
PEDI_CABE p inner join INTRASISTEMA_LOC i on (p.tipo = i.tipo)
where FECHA >= dateadd(dd,-20,getdate())
and isnull(REFERENCIA,'') <> ''
and id_negocio = @id_negocio
```

30/05 se agrego el filtro por negocio a pedido de Maxi (se toma de la tabla intrasistema_loc)

Casos de Envío - ML

Caso 1 - CARGO DE ENVIO + COSTO DE ENVIO

ID : 2000004825406917

INTRA_ID: 10867142

TOTAL_INTRA: 10707.50

TOTAL_RP: 12621.77

URL : [Detalle de operación \(mercadopago.com.ar\)](#)

Envíos ^		\$ 0,00
Costo de envío		\$ 1.914,24
Cargo por envío		- \$ 1.914,24

Caso 2 - CARGO POR ENVIO + BONIFICACIÓN

ID: 2000004825257339

INTRA_ID: 10866817

TOTAL_INTRA: 76,022.00

TOTAL_RP: 76782.99

URL: [Detalle de operación \(mercadopago.com.ar\)](#)

Envíos ^		\$ 4.819,98
Costo de envío		\$ 760,99
Cargo por envío		- \$ 760,99
Bonificación por envío		\$ 4.819,98

Caso 3 - COSTO DE ENVIO + BONICIFICACION

ID: 2000006452174542

INTRA_ID: 10864499

TOTAL_INTRA: 4.069,00

TOTAL_RP: 5996.95

URL: [Detalle de operación \(mercadopago.com.ar\)](Detalle de operación (mercadopago.com.ar))

Envíos	\$ 2.409,99
Costo de envío	\$ 1.927,99
Bonificación por envío	\$ 482,00

ANTICIPO

Revisar N° de cobro de envío para conciliación - N 4xxxxxxxx

Gestión para Pymes - ERP (Frankfull SRL.) (Usuario: full_sa)

Tablas Operaciones Consultas Ventanas Salir

Comprobantes Emitidos (Ventas)

Filtros Agrupaciones / Mas Filtros

Tipo de Comp.

Cliente []

Desde 01/07/2023 Hasta

Cód. Comprobante

1 A40 Anticipo ecommerce

Comprobante

Tipo: Anticipo ecommerce Nro.: 120376

Datos de Cabecera Totales Valores CUOTAS / Vencimientos

	Coti	Cod. Tipo Valor	Tipo	Nro.	Fecha	Banco Origen / Tarjeta	Banco Destino	Cuenta
1	1.0000	TR	Transferencia	63518287775	13/09/2023	MLP	Mercado Pago	
2	1.0000	TR	Transferencia	63518287775	13/09/2023	MLP	Mercado Pago	

Validar Fechas

Saldo Margen Gananc. -10,664.99

Pedidos - Full

(Frankfull SRL.) (Usuario: full_sa)

Tablas Ventanas Salir

U\$S

Consultas de Pedidos

Filtros

Vendedor: Desde 14/09/2023 Hasta 14/09/2023 F. Ingreso

Cliente: 9 Mercado Libre Full Desde 0 Hasta 0 OC

Nom. de Fantasia Domicilio Comprobante Pedido E-commerce WEB

Habilitar Facturación Masiva Estado Con Items Pendientes Dep. Reserva Deposito Full

	Comprobante	Nro.	Cliente	Razón Social // Nombre Fantasia	Fecha	F. Entrega

Intrasistema + ML operativo

Documentar:

○ SP - STOCK - ML

Fullescabio -

- get lista

○ Peñaflor

- get_lista

○ 01 mercado

- tecnogo_stock_lista

Vista - intrasistema_items_stock_reserva

Se usa para obtener el reservado por producto por deposito

```
SELECT ac.COD_ARTI_COMPO,p.DEPO_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
FROM factu_dba_norte.dbo.PEDI_ITEMS as p
-- inner join #art_pen ap on p.ARTICULO=ap.articulo collate database_default
inner join factu_dba_norte.dbo.ARTICULOS_COMPO as ac on p.ARTICULO=ac.COD_ARTICULO collate
database_default
WHERE DEPO_RESER in ('DEP','PEN','VYS','FUL') and p.CANT_PEND > 0 and RESER = 'S'
group by ac.COD_ARTI_COMPO,p.DEPO_RESER
```

```
create table #art_pen(articulo varchar(60) collate database_default, penaflor char(1) collate
database_default)
create unique index ind1 on #art_pen(articulo);

create table #art_comp_pen(articulo varchar(60) collate database_default,depo varchar(5) collate
database_default,reserva int)
create index ind1 on #art_comp_pen(articulo);

create table #art_res_pen(articulo varchar(60) collate database_default,depo varchar(5) collate
database_default,reserva int)
create index ind1 on #art_res_pen(articulo);

create table #art_art_pen(articulo varchar(60) collate database_default,depo varchar(5) collate
database_default,reserva int)
create index ind1 on #art_art_pen(articulo);

create table #art_fis_pen(articulo varchar(60) collate database_default,stock int)
create index ind1 on #art_fis_pen(articulo);

insert into #art_pen(articulo,penaflor)
select COD_ARTICULO , 'S'
from factu_dba_norte.dbo.ARTICULOS
where upper(AGRU_2) in ('CINBA' , 'PEÑAF' , 'TRAPI' , 'POPPÑ' )
and isnull(PAUSAR_SINCRO_MKT,'N') = 'N'
union
select p.ARTICULO , 'S'
from factu_dba_norte.dbo.PROVE_ARTI p inner join ARTICULOS a on (p.ARTICULO = a.COD_ARTICULO )
where p.PROVE = '46'
and isnull(a.PAUSAR_SINCRO_MKT,'N') = 'N'
```

```

if @deposito = 'DEP'
begin
    insert into #art_pen(articulo,penaflor)
    select COD_ARTICULO , 'N'
    from factu_dba_norte.dbo.ARTICULOS a
    where isnull(PAUSAR_SINCRO_MKT,'N') = 'N'
        and not exists (select 1 from #art_pen t where t.articulo = a.COD_ARTICULO )
    delete #art_pen where penaflor = 'S'
end

insert into #art_comp_pen(articulo,depo,reserva)
SELECT ac.COD_ARTI_COMPO,p.DEPO_RESER,sum(ac.CANTI * p.CANT_PEND) as reservada
FROM factu_dba_norte.dbo.PEDI_ITEMS as p
inner join #art_pen ap on p.ARTICULO=ap.articulo collate database_default
inner join factu_dba_norte.dbo.ARTICULOS_COMPO as ac on p.ARTICULO=ac.COD_ARTIculo collate
database_default
WHERE DEPO_RESER = @deposito and p.CANT_PEND > 0 and RESER = 'S'
group by ac.COD_ARTI_COMPO,p.DEPO_RESER

insert into #art_res_pen(articulo,depo,reserva)
SELECT p.ARTICULO, DEPO_RESER, SUM(CANT_PEND_RESER) AS RESERVADA
FROM factu_dba_norte.dbo.PEDI_ITEMS p
inner join #art_pen ap on p.ARTICULO=ap.articulo collate database_default
WHERE RESER = 'S' AND CANT_PEND_RESER > 0 and DEPO_RESER = @deposito
GROUP BY p.ARTICULO, DEPO_RESER

insert into #art_art_pen(articulo,depo,reserva)

SELECT pit.ARTICULO, DEPO_RESER, SUM(CANT_PEND) AS RESERVADA
FROM factu_dba_norte.dbo.PEDI_ITEMS pit
inner join #art_pen ap on pit.ARTICULO=ap.articulo collate database_default
WHERE RESER = 'S' AND CANT_PEND > 0 and DEPO_RESER = @deposito
GROUP BY pit.ARTICULO, DEPO_RESER

insert into #art_fis_pen(articulo,stock)
SELECT
ART.COD_ARTICULO,
DEP.CANT_STOCK as fisico
FROM factu_dba_norte.dbo.ARTICULOS AS ART
    LEFT outer JOIN factu_dba_norte.dbo.ARTICULOS_STOCK_DEPO AS DEP
    ON ART.COD_ARTICULO = DEP.COD_ARTICULO  collate database_default
        inner join #art_pen ap on art.COD_ARTICULO=ap.articulo collate database_default
WHERE ART.ACTIVO = 'S'
AND DEP.DEPO =  @deposito

insert into #art_fis_pen(articulo,stock)
SELECT

```

```

ART.COD_ARTICULO,
DEP.CANT_STOCK as fisico
FROM factu_dba_norte.dbo.ARTICULOS AS ART
    LEFT outer JOIN factu_dba_norte.dbo.ARTICULOS_STOCK_DEPO AS DEP
    ON ART.COD_ARTICULO = DEP.COD_ARTICULO collate database_default
    left join #art_pen ap on art.COD_ARTICULO=ap.articulo collate database_default
WHERE ART.ACTIVO = 'S'
AND DEP.DEPO = @deposito and ap.articulo is null

select art.COD_ARTICULO item_id,ISNULL(apc.reserva,0) +
    ISNULL(atp.reserva,0)+ISNULL(arp.reserva,0) reservado,
    isnull(afp.stock,0) fisico,
    isnull(afp.stock,0) - (ISNULL(apc.reserva,0)+ISNULL(atp.reserva,0)+ISNULL(arp.reserva,0))
disponible
    --,art.DESCRIP_ARTI
from ARTICULOS art
    inner join #art_pen ap on art.COD_ARTICULO collate database_default =ap.articulo
    left join #art_comp_pen apc on art.COD_ARTICULO collate database_default =apc.articulo
    left join #art_art_pen atp on art.COD_ARTICULO collate database_default =atp.articulo
    left join #art_res_pen arp on art.COD_ARTICULO collate database_default =arp.articulo
    left join #art_fis_pen afp on art.COD_ARTICULO collate database_default =afp.articulo

```

Vista - intrasistema_items_stock_fisico

```
ALTER view [dbo].[intrasistema_items_stock_fisico]
as
SELECT
ART.COD_ARTICULO,
dep.DEP0 ,
DEP.CANT_STOCK as fisico
FROM factu_dba_norte.dbo.ARTICULOS AS ART
    LEFT outer JOIN factu_dba_norte.dbo.ARTICULOS_STOCK_DEPO AS DEP
    ON ART.COD_ARTICULO = DEP.COD_ARTICULO    collate database_default
WHERE ART.ACTIVO = 'S'
AND DEP.DEP0 in (SELECT DEPO_CODI FROM intrasistema_items_stock_depositos WHERE ENVIAR_IS = 'S')
GO
```

Vista - intrasistema_items_stock_disponible

```
ALTER view [dbo].[intrasistema_items_stock_disponible]
as
select t.cod_articulo, t.depo, sum(t.qty) as Qty_Disponible
from (
select
f.COD_ARTICULO,
f.DEP0,
f.fisico as qty
from intrasistema_items_stock_fisico as f
union all
select
r.cod_articulo,
r.depo as depo,
r.reservada * -1 as qty
from intrasistema_items_stock_reserva as r) as t
group by t.cod_articulo, t.depo
```

GO

Vista - intrasistema_items_stock_depositos

Los depósitos que se consideran son

- Único
- Peñaflor
- Vinos y spirits
- Mercado libre Full

```
ALTER view [dbo].[intrasistema_items_stock_depositos]
as
select DEPO_CODI, DESCRIP,
'S' AS ENVIAR_IS
from DEPOSITOS
where DEPO_CODI in ('DEP', 'PEN', 'VYS', 'DML')
```

```
GO
```

Vista - intrasistema_items_stock_articulos

```
ALTER view [dbo].[intrasistema_items_stock_articulos]
as
select distinct cod_articulo, art_penaflor from (
select COD_ARTICULO , 'S' as ART_PENAFLOR
from factu_dba_norte.dbo.ARTICULOS
where upper(AGRU_2) in ('CINBA', 'PEÑAF', 'TRAPI', 'POPPÑ')
and isnull(PAUSAR_SINCRO_MKT, 'N') = 'N'
union
select p.ARTICULO , 'S' as ART_PENAFLOR
from factu_dba_norte.dbo.PROVE_ARTI p inner join ARTICULOS a
on p.ARTICULO = a.COD_ARTICULO
and p.PROVE_PRINCIPAL = 'S'
where p.PROVE = '46'
and isnull(a.PAUSAR_SINCRO_MKT, 'N') = 'N'
union
select COD_ARTICULO , 'N' as ART_PENAFLOR
from factu_dba_norte.dbo.ARTICULOS a
where isnull(PAUSAR_SINCRO_MKT, 'N') = 'N'
and COD_ARTICULO not in (
    select COD_ARTICULO
    from factu_dba_norte.dbo.ARTICULOS
    where upper(AGRU_2) in ('CINBA', 'PEÑAF', 'TRAPI', 'POPPÑ')
    and isnull(PAUSAR_SINCRO_MKT, 'N') = 'N'
    union
    select p.ARTICULO
    from factu_dba_norte.dbo.PROVE_ARTI p inner join ARTICULOS a
    on p.ARTICULO = a.COD_ARTICULO
    and p.PROVE_PRINCIPAL = 'S'
    where p.PROVE = '46'
    and isnull(a.PAUSAR_SINCRO_MKT, 'N') = 'N'
)
) as T
GO
```

Vista - intrasistema_items_stock_art_depo

```
ALTER view [dbo].[intrasistema_items_stock_art_depo]
as
select COD_ARTICULO, ART_PENAFLOR, DEPO_CODI
from  intrasistema_items_stock_articulos AS A, intrasistema_items_stock_depositos AS D
GO
```

Tablero - Reporte

SQL

dbo.stock_quiebre_vista (**factu_dba_norte**)

```
ALTER VIEW [dbo].[stock_quiebre_vista]
as
SELECT DISTINCT
A.COD_ARTICULO as CODIGO,
A.DESCRIP_ARTI AS DESCRIPCION_ARTICULO,
FLOOR(D.CANT_STOCK) AS CANT_STOCK,
FLOOR(SUM (I.CANT) OVER (PARTITION BY A.COD_ARTICULO )) AS CANT_PEDIDO
FROM PEDI_CABE AS P
INNER JOIN PEDI_ITEMS AS I ON (P.NUM = I.NUM) --Acá ya tengo la cabecera con los items
del pedido
INNER JOIN ARTICULOS AS A ON (I.ARTICULO = A.COD_ARTICULO) --Relaciono en cada item la
descripción del articulo
INNER JOIN ARTICULOS_STOCK_DEPO AS D ON (A.COD_ARTICULO = D.COD_ARTICULO) --Traigo stock
real del articulo
WHERE
I.TIPO IN ('DDD','DDI','DRF','DTE') --Pedido, pedido interno, pedido ML y pedido
tecnogo
AND P.TIPO IN ('DDD','DDI','DRF','DTE') --Pedido, pedido interno, pedido ML y pedido
tecnogo
AND P.FECHA >= GETDATE()-7 --Desde hace 7 días
AND D.DEPO = 'DEP' --Depósito dep
AND I.CANT_PEND > 0 --Sin facturar
GO
```

Nota: cuenta todos los pedidos de 7 días atrás, sumando las cantidades pendientes por producto y conciliando con el deposito.

dbo.tablero_pedidos

```
USE [factu_dba_norte]
GO
***** Object: StoredProcedure [dbo].[tablero_pedidos] Script Date: 22/09/2023 12:08:55 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
ALTER proc [dbo].[tablero_pedidos]
as
set nocount on
```

```

declare @horario datetime, @horario_ayer datetime
select @horario=SMALLDATETIMEFROMPARTS(YEAR(GETDATE()), MONTH(GETDATE()), DAY(GETDATE()), 11, 00)
select @horario_ayer=DATEADD(hour,-24,@horario)

select
    co.DESCRIP DESCRIPCION,
    c.tipo TIPO,
    c.num NUM,
    cl.razon RAZON_SOCIAL,
    cl.NOM_FANTASIA NOMBRE_FANTASIA,
    c.FECHA FECHA_PEDIDO,
    convert(varchar,c.fecha_alta,108) HORA_PEDIDO
from pedi_cabe c
inner join comp co on (c.tipo = co.COD_COMP )
inner join clientes cl on (c.CLIENTE = cl.NUM_CLIENTE)
where not exists (select 1 from RENG_FAC r where r.PEDI_TIPO=c.tipo and r.PEDI_NUM=c.num)
and c.tipo in ('DDD','DDW','DDI')
and c.FECHA >= convert(varchar,getdate() - 7,23)
order by DESCRIPCION, TIPO, NUM, FECHA_PEDIDO, HORA_PEDIDO ASC

```

Nota:

dbo.tablero_pedidos_pendientes

```

USE [factu_dba_norte]
GO
/***** Object:  StoredProcedure [dbo].[tablero_pedidos_pendiente]   Script Date: 22/09/2023 12:12:51
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[tablero_pedidos_pendiente]
as

declare @fecha datetime
select @fecha=DATEADD(hour, DATEDIFF(hour, '', DATEADD(DAY, -1, getdate()))), ''

/*
Campos a informar (solo de los pendientes)
Tipo / Comprobante / Numero / Razon Social / Nombre Fantasia / Fecha Pedido / Hora Pedido
*/

```

```

select p.TIPO TIPO,
       p.NUM    NUM,
       co.descrip DESCRIPCION,
       --cl.razon RAZON_SOCIAL ,
       cl.NOM_FANTASIA NOMBRE_FANTASIA,
       p.obs, -- se le agrega el campo obs
       convert(varchar,p.fecha,103) FECHA_PEDIDO,
       convert(varchar,p.fecha_alta,108) HORA_PEDIDO
  from pedi_cabe p
 inner join comp co on (p.tipo = co.COD_COMP )
 inner join clientes cl on (p.cliente = cl.NUM_CLIENTE)
where p.fecha between DATEADD(DAY, -7, @fecha) and @fecha
and not exists (select 1 from reng_fac rf where rf.PEDI_TIPO=p.TIPO and rf.pedi_num = p.num)
and p.tipo in ('DDD','DDW','DDI')
order by P.TIPO, P.NUM, FECHA_PEDIDO, HORA_PEDIDO

```

Nota:

dbo.tablero_comprobantes_sin_hr

```

USE [factu_dba_norte]
GO
***** Object:  StoredProcedure [dbo].[tablero_comprobantes_sin_hr]      Script Date: 22/09/2023 12:14:24
*****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[tablero_comprobantes_sin_hr]
as

set nocount on
declare @desde datetime = null, @hasta datetime = null

-- Para parametros sin valores, se pone por defecto, rango de una semana
if isnull(@desde,'')=' select @desde = convert(varchar,DATEADD(DAY,
-7,SMALLDATETIMEFROMPARTS(YEAR(GETDATE()), MONTH(GETDATE()), DAY(GETDATE()), 00,00)),112)
if isnull(@hasta,'')=' select @hasta = convert(varchar,SMALLDATETIMEFROMPARTS(YEAR(GETDATE()),
MONTH(GETDATE()), DAY(GETDATE()), 00,00 ),112)

-- Fecha/Tipo/Comprobante/Numero/CodigoCliente/RazonSocial/NombreFantasia

select convert(varchar,ce.FECHA,103) FECHA,
       c.DESCRIP COMPROBANTE,
       ce.tipo TIPO,
       ce.num NUM,
       ce.cliente CODIGO_CLIENTE,
       cl.RAZON RAZON_SOCIAL,

```

```
cl.NOM_FANTASIA NOMBRE_FANTASIA
from COMP_EMITIDOS ce
inner join comp c on (c.COD_COMP = ce.tipo)
inner join clientes cl on (ce.cliente = cl.num_cliente)
where ce.estado != 'ANU'
and ce.fecha between convert(varchar,@desde,112) and convert(varchar,@hasta,112)
and ce.TIPO in ('FAI', 'FBI' , 'FBX')
and not exists (select 1 from HOJA_RUTA_CABE c inner join HOJA_RUTA_ITEMS i on (c.NUM = i.NUM)
where i.COMPRO_TIPO = ce.tipo and i.compro_num = ce.num )
--group by ce.FECHA
ORDER BY CE.FECHA DESC, CE.TIPO, CE.NUM
```

Nota:

Integraciones con Proveedores

Chandon

SP Stock

Nombre de SP □ Base central □ full_Proveedores_stock_chandon

```
select '23804',
       ISNULL (P.CODI_PROVE, a.COD_ARTICULO) ,

       convert(varchar, convert(int,sum( isnnull(d.cant_stock,0) ))) ,
       convert(varchar(6),getdate(),112) as campo_expor
  from ARTICULOS a left join ARTICULOS_STOCK_DEPO d
    on (a.COD_ARTICULO = d.COD_ARTICULO)
   LEFT JOIN PROVE_ARTI AS P
     ON P.ARTICULO = a.COD_ARTICULO
     AND P.PROVE_PRINCIPAL = 'S'
 where exists (select 1 from PROVE_ARTI p where p.ARTICULO = a.COD_ARTICULO and p.PROVE = 17)
   and a.sin_stock = 'N'
   AND A.USA_COMPO = 'N'
   and d.DEPO not in ('MAY')
 group by ISNULL (P.CODI_PROVE, a.COD_ARTICULO)
```

- Se limita a los productos que tienen el código de proveedor 17 (chandon) como proveedor principal

```
select 1 from PROVE_ARTI p where p.ARTICULO = a.COD_ARTICULO and p.PROVE = 17
```

- No se mandan los productos que usan componentes □

```
AND A.USA_COMPO = 'N'
```

- Excluimos el stock del deposito MAYORISTA

```
and d.DEPO not in ('MAY')
```

SQL del SP

```
select '23804' + ';' +
       ISNULL (P.CODI_PROVE, a.COD_ARTICULO) +' ;' +
       convert(varchar, convert(int,sum( isnnull(d.cant_stock,0) ))) +' ;' +
       convert(varchar(6),getdate(),112) as campo_expor
  from ARTICULOS a left join ARTICULOS_STOCK_DEPO d
    on (a.COD_ARTICULO = d.COD_ARTICULO)
   LEFT JOIN PROVE_ARTI AS P
     ON P.ARTICULO = a.COD_ARTICULO
```

```
        AND P.PROVE_PRINCIPAL = 'S'  
    where exists (select 1 from PROVE_ARTI p where p.ARTICULO = a.COD_ARTICULO and p.PROVE = 17)  
        and a.sin_stock = 'N'  
        AND A.USA_COMPO = 'N'  
    group by ISNULL (P.CODI_PROVE, a.COD_ARTICULO)
```

SP Clientes

Nombre de SP □ Base central □ full_Proveedores_clientes_chandon

```
select ' COD_DISTRIBUIDOR;VENDEDOR;COD_VENDEDOR' as campo_expor
union
select DISTINCT '23804' + ',' +
c.RAZON_SOCIAL_CLIENTE + ';' +
convert(varchar,c.COD_CLIENTE) as campo_expor
from full_ventas_proveedores_vista c
where c.fecha between CONVERT(varchar, DATEADD(month, DATEDIFF(month, -1, getdate()) -1 , 0),112) and
CONVERT(varchar,DATEADD(DD,-1,DATEADD(MM,0,GETDATE()))),112)
--CONVERT(date,DATEADD(DD,-1,DATEADD(MM,0,GETDATE()))),3)
and c.proveedor = 17
--and exists (select 1 from PROVE_ARTI p where p.ARTICULO = c.COD_ARTICULO and p.PROVE = 17)
GO
```

Usa la vista □ full_ventas_proveedores_vista

SP Ventas

```
select ,
COD_DISTRIBUIDOR;FECHA;CLIENTE;RAZON_SOCIAL;NOMBRE_FANTASIA;DIRECCION;COD_POSTAL;LOCALIDAD;PROVINCIA;MATERIAL;CANTIDAD;RUBRO;CUIT;COD_VENDEDOR' campo_expor

union

select '23804' + ';' +
convert(varchar,c.fecha_datetime,112) + ';' +
convert(varchar,c.COD_CLIENTE) + ';' +
c.RAZON_SOCIAL_CLIENTE + ';' +
c.NOMBRE_CLIENTE + ';' +

isnull(c.domicilio,'') + ';' +
isnull(c.CODIGO_POSTAL,'') + ';' +
isnull(c.localidad,'') + ';' +
isnull(c.provincia,'') + ';' +
c.COD_ARTICULO + ';' +

convert(varchar, convert(int, c.CANTIDAD )) + ';' +
isnull(c.MARKETPLACE,'') + ';' +
isnull(c.ID_TRIBUTARIA,'') + ';' +
convert(varchar,c.COD_CLIENTE) campo_expor

from full_ventas_proveedores_vista c -- inner join comp cc on (c.TIPO = cc.COD_COMP )
where c.fecha between CONVERT(varchar, DATEADD(month, DATEDIFF(month, -1, getdate()) - 1, 0),112) and
CONVERT(varchar,DATEADD(DD,-1,DATEADD(MM,0,GETDATE())),112)
and c.PROVEEDOR = 17
AND DEPO_FACTURA NOT IN ('MAY')
```

Limites

Toma las ventas en un tiempo ☺ no me acuedo cual
del proveedor 17

Las ventas estan en la vista full_ventas_proveedores_vista

Vista full_ventas_proveedores_vista

```
SELECT

-- CAMBIO DEL CAMPO CLIENTE SEGUN LA AGRUPACION 3
CASE WHEN CLI.AGRU_3 = 'FULL' THEN COM.CLIENTE
     WHEN CLI.AGRU_3 = 'PRO'   THEN COM.CLIENTE
     WHEN CLI.AGRU_3 = 'VINO'  THEN COM.CLIENTE
     WHEN CLI.AGRU_3 = 'MELI'  THEN 8 --COM.CLIENTE
     ELSE '50' END AS COD_CLIENTE,

-- CODIGO DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL

COM.CLIENTE as COD_CLIENTE_ORIGINAL,

-- CAMBIO DE NOMBRE DE CLIENTE SEGUN LA AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'PRO'   THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'VINO'  THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'MELI'  THEN 'MERCADO LIBRE'
     ELSE 'e-Commerce' END AS NOMBRE_CLIENTE,

-- NOMBRE DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL

CLI.NOM_FANTASIA AS NOMBRE_CLIENTE_ORIGINAL,

-- ID TRIBUTARIA SEGUIN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN 'C'
     WHEN CLI.AGRU_3 = 'PRO'   THEN 'C'
     WHEN CLI.AGRU_3 = 'VINO'  THEN 'C'
     WHEN CLI.AGRU_3 = 'MELI'  THEN 'D'
     ELSE 'D' END AS TIPO_ID_TRIBUTARIA,

-- CUIT DEL CLIENTE O CLIENTE FINAL SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.CUIT
     WHEN CLI.AGRU_3 = 'PRO'   THEN CLI.CUIT
     WHEN CLI.AGRU_3 = 'VINO'  THEN CLI.CUIT
     WHEN CLI.AGRU_3 = 'MELI'  THEN 'CLIENTE_FINAL'
     ELSE 'CLIENTE_FINAL' END AS ID_TRIBUTARIA,
```

```

-- RAZON SOCIAL DEL CLIENTE SEGUIN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'MELI' THEN 'MERCADO LIBRE'
    ELSE 'e-Commerce' END AS RAZON_SOCIAL_CLIENTE,

-- RAZON SOCIAL ORIGINAL -> NO PASAR NUNCA, ES CONTROL

CLI.RAZON AS RAZON_SOCIAL_ORIGINAL,

-- PROVINCIA SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'MELI' THEN 'Buenos Aires'
    ELSE 'Buenos Aires' END AS PROVINCIA,

-- LOCALIDAD SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'MELI' THEN 'El Talar'
    ELSE 'El Talar' END AS LOCALIDAD,

-- DOMICILIO SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'MELI' THEN 'Colectora Panamericana 31966'
    ELSE 'Colectora Panamericana 31966' END AS DOMICILIO,

-- CODIGO POSTAL SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'MELI' THEN '1618'
    ELSE '1618' END AS CODIGO_POSTAL,

```

CLI.AGRU_3 AS AGRUPACION,
AGR_CLI_DESCRIP_AGRU AS NOMBRE_AGRUPACION,
CONCAT (COM.TIPO, '-', RIGHT (CONCAT ('0000000000', COM.NUM), 8)) AS FACTURA_NRO,
CONVERT (DATE, COM.FECHA, 13) as FECHA,
MONTH (COM.FECHA) as MES,

-- TIPO DE ARTICULO SEGUN LETRA DE COMIENZO
ISNULL (PROV_ART.CODI_PROVE, ITM.ARTICULO) AS COD_ARTICULO,
ITM.ARTICULO AS COD_ARTICULO_RP,

```

ART.DESCRIP_ARTI AS DESCRIPCION,
ART.COD_BARRAS AS EAN,
CASE WHEN SUBSTRING (ITM.ARTICULO,1,1) = 'C' THEN 'CENTRAL'
      WHEN SUBSTRING (ITM.ARTICULO,1,2) = 'PR' THEN 'PROMOCION'
      WHEN SUBSTRING (ITM.ARTICULO,1,1) = 'B' THEN 'COMBO'
      WHEN SUBSTRING (ITM.ARTICULO,1,1) = 'P' THEN 'PUBLICITARIO'
      ELSE '' END AS TIPO_DE_ARTICULO,

-- SI EL PRODUCTO TIENE COMPONENTES. NO TIENE CANTIDAD, PORQUE SE DESCUENTA DE LOS PRODUCTOS
ITM.CANT * -1 AS CANTIDAD,
--CASE WHEN ART.USA_COMPO = 'S' THEN 1
--      ELSE COM.CANT * -1 END AS CANTIDAD,
ITM.CANT * ITM.PRECIO_NETO * -1 AS IMPORTE_S_IVA,
SUBSTRING(COM.TIPO,1,1) AS TIPO_FACTURA,


CASE WHEN CLI.AGRU_3 = 'FULL' THEN 'FULL'
      WHEN CLI.AGRU_3 = 'PRO' THEN 'FULL'
      WHEN CLI.AGRU_3 = 'VINO' THEN 'BLANCA'
      WHEN CLI.AGRU_3 = 'MELI' THEN 'MERCADO LIBRE'
      ELSE 'e-Commerce' END AS MARKETPLACE,


-- SE TIENE QUE FILTRAR POR PROVEEDOR, ARMAR EL CONTROL DE PRODUCTOS SIN PROVEEDOR

PROV_ART.PROVE AS PROVEEDOR,
ART.USA_COMPO,
COM.fecha fecha_datetime ,
TIPO_COMP.COD_COMP ,
COM.CODI_VENDE ,
TIPO_COMP.COD_GRUPO ,
ART.AGRU_1 ,
ART.SIN_STOCK ,
ART.UM_CONVER_COMPRA,
ITM.RENGLON,
ITM.DEP0 AS DEPO_FACTURA

FROM dbo.RENG_FAC AS ITM LEFT JOIN dbo.COMP_EMITIDOS AS COM
ON COM.NUM = ITM.NUM_FACT AND COM.TIPO = ITM.TIPO_FACT

LEFT JOIN dbo.ARTICULOS AS ART
ON ITM.ARTICULO = ART.COD_ARTICULO

LEFT JOIN dbo.CLIENTES AS CLI
ON COM.CLIENTE = CLI.NUM_CLIENTE

LEFT JOIN dbo.COMP AS TIPO_COMP
ON ITM.TIPO_FACT = TIPO_COMP.COD_COMP

LEFT JOIN dbo.PROVE_ARTI AS PROV_ART

```

```

ON PROV_ART.ARTICULO = ITM.ARTICULO AND PROVE_PRINCIPAL = 'S'

LEFT JOIN dbo.AGRUP_CLIENTES AS AGR_CLI
ON CLI.AGRU_3 = AGR_CLI.CODI_AGRU AND AGR_CLI.NUM_AGRU = 3

WHERE COM.ESTADO != 'ANU'
AND LEFT(COM.TIPO,1) in ('F','C','B')
AND COM.CLIENTE <> '7066'
and SUBSTRING (ITM.ARTICULO,1,1) IN ('C','P','B')
and ( COM.actua_stock = 'S' or ITM.val_tipo is not null)

UNION ALL

/* SE SUMA LA PARTE DE MAYORISTA TRANSFORMADA */

SELECT

-- CAMBIO DEL CAMPO CLIENTE SEGUN LA AGRUPACION 3
CASE WHEN CLI.AGRU_3 = 'FULL' THEN DRW.CLIENTE
     WHEN CLI.AGRU_3 = 'PRO' THEN DRW.CLIENTE
     WHEN CLI.AGRU_3 = 'VINO' THEN DRW.CLIENTE
     WHEN CLI.AGRU_3 = 'MELI' THEN 8 --COM.CLIENTE
     ELSE '50' END AS COD_CLIENTE,

-- CODIGO DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL

DRW.CLIENTE as COD_CLIENTE_ORIGINAL,

-- CAMBIO DE NOMBRE DE CLIENTE SEGUN LA AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'PRO' THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'VINO' THEN CLI.NOM_FANTASIA
     WHEN CLI.AGRU_3 = 'MELI' THEN 'MERCADO LIBRE'
     ELSE 'e-Commerce' END AS NOMBRE_CLIENTE,

-- NOMBRE DE CLIENTE ORIGINAL -> NO PASAR NUNCA, ES CONTROL

CLI.NOM_FANTASIA AS NOMBRE_CLIENTE_ORIGINAL,

-- ID TRIBUTARIA SEGUIN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN 'C'
     WHEN CLI.AGRU_3 = 'PRO' THEN 'C'
     WHEN CLI.AGRU_3 = 'VINO' THEN 'C'
     WHEN CLI.AGRU_3 = 'MELI' THEN 'D'
     ELSE 'D' END AS TIPO_ID_TRIBUTARIA,

-- CUIT DEL CLIENTE O CLIENTE FINAL SEGUN AGRUPACION 3

```

```

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.CUIT
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.CUIT
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.CUIT
    WHEN CLI.AGRU_3 = 'MELI' THEN 'CLIENTE FINAL'
    ELSE 'CLIENTE_FINAL' END AS ID_TRIBUTARIA,

-- RAZON SOCIAL DEL CLIENTE SEGUIN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.RAZON
    WHEN CLI.AGRU_3 = 'MELI' THEN 'MERCADO LIBRE'
    ELSE 'e-Commerce' END AS RAZON_SOCIAL_CLIENTE,

-- RAZON SOCIAL ORIGINAL -> NO PASAR NUNCA, ES CONTROL

CLI.RAZON AS RAZON_SOCIAL_ORIGINAL,

-- PROVINCIA SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.PROVINCIA
    WHEN CLI.AGRU_3 = 'MELI' THEN 'Buenos Aires'
    ELSE 'Buenos Aires' END AS PROVINCIA,

-- LOCALIDAD SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.LOCALIDAD
    WHEN CLI.AGRU_3 = 'MELI' THEN 'El Talar'
    ELSE 'El Talar' END AS LOCALIDAD,

-- DOMICILIO SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.DOMICILIO
    WHEN CLI.AGRU_3 = 'MELI' THEN 'Colectora Panamericana 31966'
    ELSE 'Colectora Panamericana 31966' END AS DOMICILIO,

-- CODIGO POSTAL SEGUN AGRUPACION 3

CASE WHEN CLI.AGRU_3 = 'FULL' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'PRO' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'VINO' THEN CLI.CP
    WHEN CLI.AGRU_3 = 'MELI' THEN '1618'
    ELSE '1618' END AS CODIGO_POSTAL,

CLI.AGRU_3 AS AGRUPACION,
AGR_CLI.DESCRIP_AGRU AS NOMBRE_AGRUPACION,
CONCAT(DRW.TIPO_FACT , '-' , RIGHT(CONCAT('0000000000' , DRW.NUM_FACT) , 8)) AS FACTURA_NRO,

```

```

    CONVERT (DATE, DRW.FECHA, 13) as FECHA,
    MONTH (DRW.FECHA) as MES,

    -- TIPO DE ARTICULO SEGUN LETRA DE COMIENZO
    ISNULL (PROV_ART.CODI_PROVE, DRW.ARTICULO ) AS COD_ARTICULO,
    DRW.ARTICULO AS COD_ARTICULO_RP,

    ART.DESCRIP_ARTI AS DESCRIPCION,
    ART.COD_BARRAS AS EAN,
    CASE WHEN SUBSTRING (DRW.ARTICULO,1,1) = 'C' THEN 'CENTRAL'
        WHEN SUBSTRING (DRW.ARTICULO,1,2) = 'PR' THEN 'PROMOCION'
        WHEN SUBSTRING (DRW.ARTICULO,1,1) = 'B' THEN 'COMBO'
        WHEN SUBSTRING (DRW.ARTICULO,1,1) = 'P' THEN 'PUBLICITARIO'
        ELSE '' END AS TIPO_DE_ARTICULO,

    -- SI EL PRODUCTO TIENE COMPONENTES. NO TIENE CANTIDAD, PORQUE SE DESCUENTA DE LOS PRODUCTOS
    DRW.CANT * 1 AS CANTIDAD,
    --CASE WHEN ART.USA_COMPO = 'S' THEN 1
    --     ELSE COM.CANT * -1 END AS CANTIDAD,
    DRW.CANT * DRW.PRECIO_NETO * 1 AS IMPORTE_S_IVA,
    SUBSTRING(DRW.TIPO_FACT ,1,1) AS TIPO_FACTURA,


    CASE WHEN CLI.AGRU_3 = 'FULL' THEN 'FULL'
        WHEN CLI.AGRU_3 = 'PRO' THEN 'FULL'
        WHEN CLI.AGRU_3 = 'VINO' THEN 'BLANCA'
        WHEN CLI.AGRU_3 = 'MELI' THEN 'MERCADO LIBRE'
        ELSE 'e-Commerce' END AS MARKETPLACE,

    -- SE TIENE QUE FILTRAR POR PROVEEDOR, ARMAR EL CONTROL DE PRODUCTOS SIN PROVEEDOR

    PROV_ART.PROVE AS PROVEEDOR,
    ART.USA_COMPO,
    DRW.fecha fecha_datetime ,
    TIPO_COMP.COD_COMP ,
    '' AS CODI_VENDE ,
    TIPO_COMP.COD_GRUPO ,
    ART.AGRU_1 ,
    ART.SIN_STOCK ,
    ART.UM_CONVER_COMPRA,
    DRW.RENGLON,
    DRW.DEP0 AS DEPO_FACTURA

FROM dbo.RENG_FAC_DRAW AS DRW LEFT JOIN dbo.ARTICULOS AS ART
    ON DRW.ARTICULO = ART.COD_ARTICULO

    LEFT JOIN dbo.CLIENTES AS CLI
    ON DRW.CLIENTE = CLI.NUM_CLIENTE

    LEFT JOIN dbo.COMP AS TIPO_COMP

```

```
ON DRW.TIPO_FACT = TIPO_COMP.COD_COMP  
  
LEFT JOIN dbo.PROVE_ARTI AS PROV_ART  
ON PROV_ART.ARTICULO = DRW.ARTICULO AND PROVE_PRINCIPAL = 'S'  
  
LEFT JOIN dbo.AGRUP_CLIENTES AS AGR_CLI  
ON CLI.AGRU_3 = AGR_CLI.CODI_AGRU AND AGR_CLI.NUM_AGRU = 3  
  
WHERE LEFT(DRW.TIPO_FACT ,1) in ('F','C','B')  
AND DRW.CLIENTE <> '7066'  
and SUBSTRING (DRW.ARTICULO,1,1) IN ('C','P','B')
```

28-02-2024 ☐ SE SUMO LA SEGUNDA PARTE CON LA TABLA DRW

DBA Particulares

Consulta la base de clientes, con datos puntuales que necesitan. En este caso, consulta los clientes con sus vendedores

Tanto para DBA Norte como para DBA Centro

Utiliza : Cecilia - DBA Norte

Vista : reporte_cliente

Base : Centro y Norte

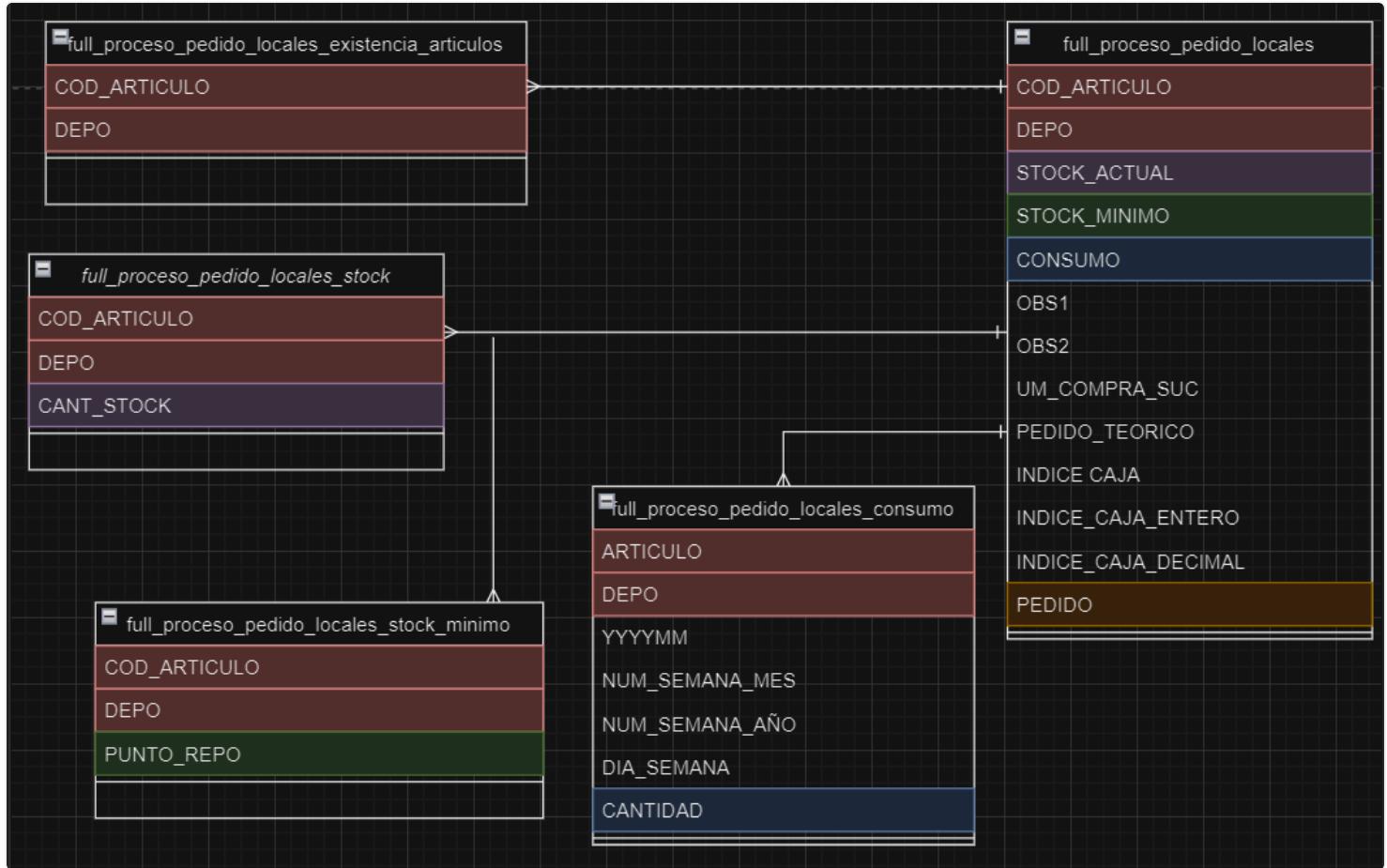
File:

[Reporte_cliente.xlsx](#)

```
factu_dba_norte
select
c.NUM_CLIENTE,
c.RAZON,
c.NOM_FANTASIA,
c.LOCALIDAD,
v.nombre as VENDEDOR,
c.MAIL,
c.AGRU_1,
A.DESCRIP_AGRU,
C.ESTADO,
C.IMPOR_LIMI_CREDI,
C.CONDI_VENTA
from CLIENTES as C inner join AGRUP_CLIENTES as a on ( C.AGRU_1 = A.CODI_AGRU)
inner join VENDEDORES as v on (c.CODI_VENDE = v.CODI_VENDE)
```

Pedidos sucursales propias

Diagrama



[diagrama.drawio](#)

Restricciones

Restricciones:

- Se tienen en cuenta las sucursales propias. Restricción en Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19374>)

Pero la tabla articulos_punto_repo solo tiene los depositos DEP, BHB, PMA, FRS, DR, FLO, FRS, ROS, PAL, PIL

- Se tienen en cuenta solo productos C. Restricción en Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19354>)
- Se consideran los consumos en un rango de 6 meses atrás. Restricción en Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19314>)

full_proceso_pedido_locales

```
ALTER VIEW [dbo].[full_proceso_pedido_locales]
as
select distinct
    T.COD_ARTICULO,
    a.DESCRIP_ARTI ,
    a.PRECIO_UNI,
    a.Imp_Interno ,
    a.Precio_c_IVA,
    T.DEPO ,
    s.CANT_STOCK STOCK_ACTUAL,
    isnull(sm.PUNTO_REPO, 1 ) STOCK_MINIMO,
    CEILING (SUM(c.cantidad) / 24) CONSUMO,
    case
        when s.cant_stock < isnull(sm.punto_repo,1) then 'PE' --pedir
        when s.cant_stock = isnull(sm.punto_repo,1) then 'NA' --no actualizar
        when s.cant_stock > isnull(sm.punto_repo,1) then 'SS' --sobra stock
    end as OBS1,

    case
        --casos PE
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
        then 'PEME' --menor
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
        then 'PEIG' --igual
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
        then 'PEMA' --mayor

        --casos NA
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
        then 'NAME'
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
        then 'NAIG'
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
        then 'NAMA'

        --casos SS
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
        then 'SSME'
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
        then 'SSIG'
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
        then 'SSMA'
```

```

end as OBS2,

CASE WHEN A.UM_COMPRA_SUC > 0 THEN A.UM_COMPRA_SUC ELSE 1 END AS UM_COMPRA_SUC,

isnull(sm.punto_repo,1) - s.CANT_STOCK + CEILING (SUM(c.cantidad) / 24) as PEDIDO_TEORICO,
((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as
INDICE_CAJA,
cast((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc as
integer) INDICE_CAJA_ENTERO,
((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
INDICE_CAJA_DECIMAL,
case
    --mayor 0.60
    when ((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
> 0.60 and a.UM_Compra_Suc <> 1
        then (cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) /
a.UM_Compra_Suc) as integer)) * A.UM_COMPRA_SUC + A.UM_COMPRA_SUC
    --menor 0.60
    when ((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
< 0.60 and a.UM_Compra_Suc <> 1
        then (cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) /
a.UM_Compra_Suc) as integer)) * A.UM_COMPRA_SUC
    when a.UM_Compra_Suc = 1
        then isnull(sm.punto_repo,1) - s.CANT_STOCK + CEILING (SUM(c.cantidad) / 24)
end as PEDIDO

from full_proceso_pedido_locales_existencia_articulo as T
inner join full_proceso_pedido_locales_consumo as c
on (T.COD_ARTICULO COLLATE Modern_Spanish_CI_AS = c.articulo COLLATE Modern_Spanish_CI_AS and
T.DEP0 COLLATE Modern_Spanish_CI_AS = c.depo COLLATE Modern_Spanish_CI_AS)

```

```

inner join full_proceso_pedido_locales_stock as s
on (T.COD_ARTICULO COLLATE Modern_Spanish_CI_AS = s.COD_ARTICULO COLLATE Modern_Spanish_CI_AS
and T.DEP0 COLLATE Modern_Spanish_CI_AS = s.DEP0 COLLATE Modern_Spanish_CI_AS)

left join full_proceso_pedido_locales_stock_minimo as sm --NOTA: left join porque esta tabla solo
tiene los depos DEP, BHB, PMA, FRS, DR, FLO, FRS, ROS, PAL, PIL
on (T.COD_ARTICULO COLLATE Modern_Spanish_CI_AS = sm.COD_ARTICULO COLLATE Modern_Spanish_CI_AS
and T.DEP0 COLLATE Modern_Spanish_CI_AS = sm.DEP0 COLLATE Modern_Spanish_CI_AS)

inner JOIN full_powerbi_kpi_empresa_articulos AS a
ON C.ARTICULO COLLATE Modern_Spanish_CI_AS = A.COD_ARTICULO COLLATE Modern_Spanish_CI_AS

where t.depo = 'LHE'

GROUP BY
T.COD_ARTICULO,
T.depo,
s.cant_stock,
sm.punto_repo,
A.UM_COMPRA_SUC,
a.DESCRIP_ARTI ,
a.PRECIO_UNI,
a.Imp_Interno,
a.Precio_c_IVA

```

GO

- Se suma la cantidad.

CEILING (SUM(c.cantidad) / 24) CONSUMO

- Se evalúa el stock actual del local vs el stock que deben tener.

```

case
when s.cant_stock < isnull(sm.punto_repo,1) then 'PE' --pedir
when s.cant_stock = isnull(sm.punto_repo,1) then 'NA' --no actualizar
when s.cant_stock > isnull(sm.punto_repo,1) then 'SS' --sobra stock
end as OBS1,

```

- Se evalúa, si piden, no piden o le sobra stock por menor unidad, igual o mayor al stock actual

```

case
    --casos PE
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
    then 'PEME' --menor
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
    then 'PEIG' --igual
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock < isnull(sm.punto_repo,1)
    then 'PEMA' --mayor

    --casos NA
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
    then 'NAME'
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
    then 'NAIG'
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock = isnull(sm.punto_repo,1)
    then 'NAMA'

    --casos SS
        when CEILING (SUM(c.cantidad) / 24) < s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
    then 'SSME'
        when CEILING (SUM(c.cantidad) / 24) = s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
    then 'SSIG'
        when CEILING (SUM(c.cantidad) / 24) > s.CANT_STOCK and s.cant_stock > isnull(sm.punto_repo,1)
    then 'SSMA'

    end as OBS2,

```

- Si la unidad de compra de la sucursal es igual a 0 se le asigna 1.

```
CASE WHEN A.UM_COMPRA_SUC > 0 THEN A.UM_COMPRA_SUC ELSE 1 END AS UM_COMPRA_SUC
```

- Se hace el stock mínimo que debe tener el local - el stock actual y se le suma el consumo promedio que tiene de ese producto, en una semana, el local

```
isnull(sm.punto_repo,1) - s.CANT_STOCK + CEILING (SUM(c.cantidad) / 24) as PEDIDO_TEORICO,
```

- Se hace lo mismo solo que al consumo promedio se lo divide por la unidad de compra de la sucursal

```
((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as INDICE_CAJA,
```

- Se saca la parte entera del indice de caja y la parte decimal

```
cast((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc as integer) INDICE_CAJA_ENTERO,
```

```

((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
INDICE_CAJA_DECIMAL,

```

- Se evalúa: si la parte decimal es mayor a 0.60 y la unidad de compra es distinto a 1, se hace la parte entera * la unidad de compra + una unidad de compra más para sacar el total del pedido.

Si es menor a 0, se hace la parte entera * la unidad de compra (Es decir, sin sumar una unidad de compra más al pedido).

Si la unidad de compra es igual a uno, se hace la unidad mínima de reposición menos el stock sumado al consumo.

```

case

    --mayor 0.60
    when ((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
> 0.60 and a.UM_Compra_Suc <> 1
        then (cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer)) * A.UM_COMPRA_SUC + A.UM_COMPRA_SUC

    --menor 0.60
    when ((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) -
(cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer))
< 0.60 and a.UM_Compra_Suc <> 1
        then (cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc) as integer)) * A.UM_COMPRA_SUC

    when a.UM_Compra_Suc = 1
        then isnull(sm.punto_repo,1) - s.CANT_STOCK + CEILING (SUM(c.cantidad) / 24)
end as PEDIDO

```

NOTA: En la extracción de la parte entera y la parte decimal de indice de caja no se puede usar la función FLOOR porque redondea para arriba cuando el número supera 0.5. Lo mismo pasa con la función CAST. Pero, para que en CAST no ocurra se lo debe hacer "as integer". Ejemplo:

```
cast(((isnull(sm.punto_repo,1) - s.CANT_STOCK + (SUM(c.cantidad) / 24)) / a.UM_Compra_Suc as integer)
```

NOTA: left join a la tabla full_proceso_pedidos_locales_stock_minimo porque esta tabla solo tiene los depós DEP, BHB, PMA, FRS, DR, FLO, FRS, ROS, PAL, PIL

```
left join full_proceso_pedido_locales_stock_minimo as sm
    on (T.COD_ARTICULO COLLATE Modern_Spanish_CI_AS = sm.COD_ARTICULO COLLATE Modern_Spanish_CI_AS
and T.DEPO COLLATE Modern_Spanish_CI_AS = sm.DEPO COLLATE Modern_Spanish_CI_AS)
```

Vistas que usa

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19314>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19354>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19374>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19454>)

full_proceso_pedido_locales_consumo

```
ALTER VIEW [dbo].[full_proceso_pedido_locales_consumo]
as
SELECT
    L.ARTICULO,
    L.DEP0,
    FORMAT(CONVERT(DATE, L.FECHA_COMPLETA, 103), 'yyyy-MM') AS YYYYMM,
    DATEPART(WEEK, CONVERT(DATE, L.FECHA_COMPLETA, 103)) AS NUMERO_SEMANA_ANO,
    DATEPART(WEEK, CONVERT(DATE, L.FECHA_COMPLETA, 103)) - DATEPART(WEEK, DATEADD(MM, DATEDIFF(MM, 0,
    CONVERT(DATE, L.FECHA_COMPLETA, 103)), 0)) + 1 NUM_SEMANA_MES, --Semana
    DATEPART(WEEK, CONVERT(DATE, L.FECHA_COMPLETA, 103)) AS NUM_SEMANA_ANO,
    DATEPART(weekday, CONVERT(DATE, L.FECHA_COMPLETA, 103)) AS DIA_SEMANA,
    SUM(L.CANT) CANTIDAD --OVER(PARTITION BY d.COD_ARTICULO)) AS CANTIDAD_TOTAL_V_POR_DIA --total venta
por día
FROM factu_full_reportes.dbo.pbi_ventas_locales l

    INNER JOIN
        full_powerbi_KPI_Sucursales_Sucursales s
    on L.DEP0 COLLATE Modern_Spanish_CI_AS = s.Deposito COLLATE Modern_Spanish_CI_AS
    and s.SUCURSAL_PROPIA = 'S'

    INNER JOIN
        factu_full_reportes.dbo.pbi_calendario cl
    on cl.YYYYMM = FORMAT(CONVERT(DATE, L.FECHA_COMPLETA, 103), 'yyyy-MM')

WHERE
    L.ARTICULO != 'DTO-CUMP-ORO' --no sería necesario
    and cl.POSICION between '1' and '6'

GROUP BY
    L.ARTICULO,
    L.DEP0,
    L.FECHA_COMPLETA,
    DATEPART(WEEK, CONVERT(DATE, L.FECHA_COMPLETA, 103)) - DATEPART(WEEK, DATEADD(MM, DATEDIFF(MM, 0,
    CONVERT(DATE, L.FECHA_COMPLETA, 103)), 0)) + 1 , --Semana
    DATEPART(weekday, CONVERT(DATE, L.FECHA_COMPLETA, 103))
GO
```

Vista usada en

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19394>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19454>)

full_proceso_pedido_locales_stock

```
--use factu_full_reportes
ALTER view [dbo].[full_proceso_pedido_locales_stock]
as
select
    *
from factu_full_locales.dbo.ARTICULOS_STOCK_DEPO
where COD_ARTICULO like 'C%'
GO
```

Vista usada en

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19394>)

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19454>)

full_proceso_pedido_locales_stock_minimo

```
ALTER view [dbo].[full_proceso_pedido_locales_stock_minimo] as  
  
select r.*  
from factu_full_locales.dbo.ARTICULOS_PUNTO_REPO as R  
inner join  
full_powerbi_KPI_Sucursales_Sucursales as S  
on r.depo = s.Deposito  
and s.sucursal_propia = 'S'
```

GO

NOTA: la tabla articulos_punto_repo solo tiene los depositos DEP, BHB, PMA, FRS, DR, FLO, FRS, ROS, PAL, PIL

Vista usada en

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19394>)

full_proceso_pedido_locales_existencia_articulo

```
ALTER VIEW [dbo].[full_proceso_pedido_locales_existencia_articulo]
as

select distinct
    T.COD_ARTICULO,
    T.DEP0
from (select distinct ARTICULO COLLATE Modern_Spanish_CI_AS as COD_ARTICULO, DEPO COLLATE
Modern_Spanish_CI_AS as DEPO
        from full_proceso_pedido_locales_consumo
        union all
        select distinct COD_ARTICULO COLLATE Modern_Spanish_CI_AS as COD_ARTICULO, DEPO COLLATE
Modern_Spanish_CI_AS as DEPO
        from full_proceso_pedido_locales_stock ) as T

GO
```

Vista usada en

Private (<https://app.clickup.com/30939198/docs/xg61y-5844/xg61y-19394>)