Modelo de Reportes con Reporting Services

Documento v1.0

# Revisiones del Documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Actualización** | **Fecha** | **Autor** |
| 0.1 | Creación del documento  Capítulo requerimientos, definiciones y estándar de los reportes.  Agregar listado de los reportes de EVA a migrar. | 03/08/2018 | Adriana Palacio P. |
| 0.2 | Detalle Reporte Excepción Ventas con Bono | 06/08/2018 | Adriana Palacio P. |
| 0.3 | Detalle Reporte Venta Diaria | 08/08/2018 | Adriana Palacio P. |
| 0.4 | Detalle Reporte Acumulados por Departamento | 09/08/2018 | Adriana Palacio P. |
| 0.5 | Detalle Reporte Ventas por Artículo | 13/08/2018 | Adriana Palacio P. |
| 0.6 | Detalle Reporte Ventas Por Hora | 14/08/2018 | Adriana Palacio P. |
| 0.7 | Detalle Reporte Devoluciones por Artículo | 15/08/2018 | Adriana Palacio P. |
| 0.8 | Detalle Reporte Ventas Anuladas | 16/08/2018 | Adriana Palacio P. |
| 0.9 | Detalle Reporte Ventas a Clientes | 17/08/2018 | Adriana Palacio P. |
| 0.10 | Detalle Reporte Impuesto Diario | 21/08/2018 | Adriana Palacio P. |
| 0.11 | Detalle Reporte Consolidado de Ventas (Contabilidad Consolidada)  Detalle Reporte Informe Venta Día Localidad Usuarios (Venta por Usuario) (Localidad Usuarios Recogidas) | 23/08/2018 | Adriana Palacio P. |
| 0.12 | Detalle Reporte Cierre de Caja  Detalle Reporte Porcentaje de Participación de Artículos Vendidos | 24/08/2018 | Adriana Palacio P. |
| 0.13 | Detalle Reporte Venta por Código de Barra | 27/08/2018 | Adriana Palacio P. |
| 1.0 | Versión Lista | 28/08/2018 | Adriana Palacio P. |
| 1.1 | Agregar a los reportes el filtro de ciudad | 28/09/2018 | Adriana Palacio P. |

Los reportes se construyen tomando la base de datos EVA del servidor 192.168.30.189 que tiene información de Mercadería de 14.02.2016 hasta 08.08.2017

# Requerimiento

El presente documento muestra el modelo de reportes para Mercadería usando Reporting Services, así como el listado de los que ya se encuentran creados en la herramienta.

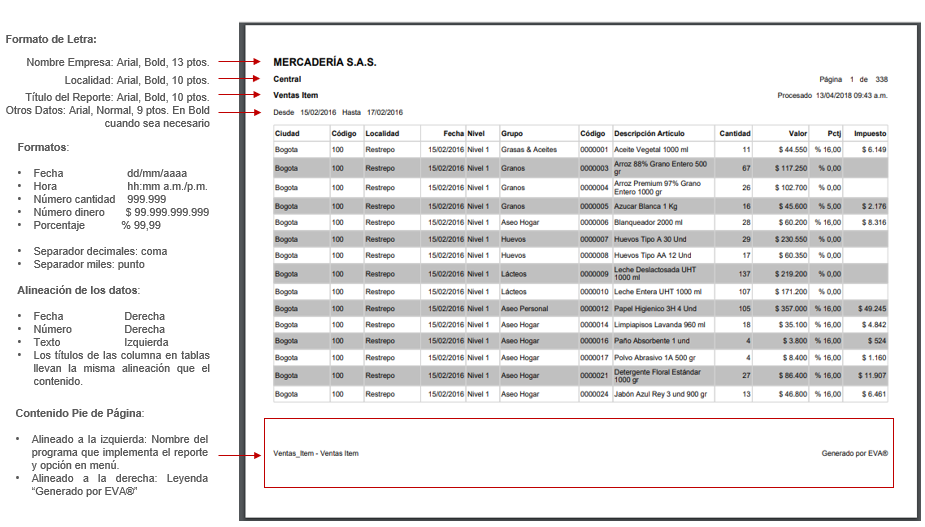
# Definiciones

* **Anulaciones**: Se refieren a las transacciones de anulaciones -tipo\_transac = 7-.
* **Devoluciones**: Se refieren a las transacciones de venta normales -tipo\_transac=0- con bruto\_pstv = 0 y bruto\_ngtv > 0 y a las transacciones de devoluciones -tipo\_transac = 20-.
* **Ventas Brutas**: Se refieren a las transacciones de venta normales -tipo\_transac=0- con bruto\_pstv <> 0-.
* **Ventas Netas**: Se refieren a las transacciones de venta normales -tipo\_transac=0- con bruto\_pstv <> 0 y las devoluciones.
* **Préstamos**: Se refieren a las transacciones de préstamos -tipo\_transac=3-.
* **Recogidas**: Se refieren a las transacciones de recogidas -tipo\_transac=4-.

# Estándar de los reportes

Las siguientes imágenes muestran el modelo de los reportes que se debe usar, incluyendo márgenes, tipo de letra, formato y alineación de los datos. Adicionalmente, se debe tener en cuenta que las abreviaturas no llevan punto y deben estar estandarizadas en todos los reportes. Para el caso de los reportes tabulares, las líneas van sombreadas de forma alterna con color “White” y “Silver.”





# Reportes

## Acumulados por Departamento

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por departamento/grupo/categoría, la cantidad, el valor bruto, valor total y el porcentaje de participación de la venta, consolidado para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_AcumuladoDpto | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Contabilidad Consolidada

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad, la cantidad de clientes y el valor de la venta bruta, la venta neta, las anulaciones, las cancelaciones, los préstamos, las recogidas, y descuentos para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_ContabilidadConsolidada | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Ventas por Hora

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por hora, la cantidad de transacciones, la cantidad de artículos vendidos, el promedio de artículos por transacción, el promedio del valor de la venta por transacción y el valor total de la venta en un rango de fechas específico y consolidado para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaPorHora | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Ventas por Artículo

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por artículo, las unidades vendidas, el valor total de la venta bruta (sin tener en cuenta devoluciones), y el valor del impuesto, en un rango de fechas específico y un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaPorArticulo | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Devoluciones Por Artículo

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por artículo, las unidades vendidas, el valor total y el valor del impuesto de las devoluciones, en un rango de fechas específico y un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_DevolucionPorArticulo | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Venta Diaria

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad y día, el valor total/promedio de la venta neta y la cantidad total/promedio de clientes en un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaDiaria | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Impuesto Diario

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad y día, el valor total/promedio de los impuestos y la cantidad total/promedio de clientes en un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_ImpuestoDiario | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Ventas Anuladas

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza, el detalle de las anulaciones en un rango de fechas específico y un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaAnulada | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Porcentaje de Participación de Artículos Vendidos

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por artículo, las unidades vendidas, el valor de la venta y el porcentaje de participación de los artículos en la venta total, consolidadas para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_ParticipacionArticulo | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Cierre de Caja

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad, terminal y día la cantidad de clientes y el valor de la venta bruta, la venta neta, las anulaciones, las devoluciones, las cancelaciones, los préstamos y las recogidas para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_CierreCaja | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Venta por Usuario

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad, terminal, día y usuario, la cantidad de clientes y el valor de la venta bruta, la venta neta, las anulaciones, las devoluciones, las cancelaciones, los préstamos y las recogidas para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaPorUsuario | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Ventas a Clientes

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por cliente y factura, el valor de la venta en un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentasCliente | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Ventas por Código de Barra

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza las unidades vendidas y el valor de la venta de los artículos que fueron vendidos por código de barra, consolidadas para un rango de fechas específico y para un conjunto de localidades dadas. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Ciudad * Proc\_Localidades * Proc\_VentaPorCodBarra | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **ciudad** | Lista de ciudades de donde se quieren obtener los datos. Ej: ‘Bogotá’, ‘Medellin’ | Lista de strings |
| **localidades** | Lista de códigos de localidades de donde se quieren obtener los datos. Ej: '100','101','102' | Lista de strings de 3 caracteres. |

## Excepción Ventas con Bono

|  |  |  |
| --- | --- | --- |
| **Descripción** | El reporte visualiza por localidad, la cantidad de bonos y valor total de la venta para aquellas ventas donde se usaron más bonos que los permitidos (1 bono por cada 20.000 pesos) en un rango de fechas específico. | |
| **Consultas SQL** | * Proc\_Empresa * Proc\_Exc\_VentasBonos | |
| **Parámetro** | **Descripción** | **Tipo** |
| **fechaInicial** | Fecha inicial del rango a seleccionar. Ej.: '01-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |
| **fechaFinal** | Fecha final del rango a seleccionar. Ej.: '04-01-2017' | Fecha  dd-mm-yyyy hh:mm:ss |

# Procedimientos

## Proc\_AcumuladoDpto

CREATE PROCEDURE [dbo].[Proc\_AcumuladoDpto]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

WITH #g AS

(

SELECT ng.descrip nivel, g.id\_grupo, g.cod\_grupo, gp.descrip

FROM grupo g

INNER JOIN grupo gp ON g.cod\_grupo = gp.cod\_grupo AND gp.subgrupo1 = '00' AND gp.subgrupo2 = '00' AND gp.subgrupo3 = '00' AND gp.subgrupo4 = '00'

INNER JOIN dbo.grupo\_nivel ng ON ng.id\_grupo\_nivel = gp.grp\_nivel\_id

)

--detalle ventas articulo

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

#g.nivel nivel,

#g.cod\_grupo AS CodGrupo,

#g.descrip AS grupo,

SUM(va.cantidad) AS cantidad,

SUM(va.valor\_venta) AS valor,

(SUM(va.valor\_venta)/

(SELECT sum(va.valor\_venta)

FROM ventas\_articulo AS va

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(va.cod\_terminal,1,3)

WHERE va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

)

) AS PartVenta

FROM ventas\_articulo AS va

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(va.cod\_terminal,1,3)

INNER JOIN articulo a ON va.id\_articulo = a.id\_articulo

LEFT JOIN #g ON a.id\_grupo = #g.id\_grupo

WHERE va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY l.cod\_localidad, l.descrip, #g.nivel, #g.cod\_grupo, #g.descrip

END

## Proc\_ContabilidadConsolidada

CREATE PROCEDURE [dbo].[Proc\_ContabilidadConsolidada]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

WITH ve AS (

SELECT v.id\_venta,

l.cod\_localidad as codigo,

l.descrip as localidad,

convert(date,v.date\_created) fecha,

case when v.cod\_tipo\_transac = '0' and v.bruto\_pstv = 0 and v.bruto\_ngtv <> 0 then '20'

else v.cod\_tipo\_transac end cod\_tipo\_transac,

vr.descrip,

case when v.cod\_tipo\_transac = '3' then isnull(sum(vmp.valor),0) else v.bruto\_pstv end pstv,

case when v.cod\_tipo\_transac = '4' then isnull(sum(vmp.valor),0) else v.bruto\_ngtv end ngtv,

case when v.cod\_tipo\_transac = '0' then isnull(sum(vmp.valor),0) else 0 end valormp

from venta v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

LEFT JOIN ventas\_recogida vr on vr.id\_venta = v.id\_venta

LEFT JOIN ventas\_medios\_pago vmp on vmp.id\_venta = v.id\_venta and (v.id\_tipo\_transac in ('3','4') or vmp.id\_medio\_pago <> '1')

WHERE @Localidades LIKE ('%' + l.cod\_localidad + '%')

AND v.date\_created between dateadd(dd,0,@fechaInicial) and dateadd(dd,1,@fechaFinal)

GROUP BY v.id\_venta, l.cod\_localidad, l.descrip, convert(date,v.date\_created), v.cod\_tipo\_transac, case when v.cod\_tipo\_transac = '0' and v.bruto\_pstv = 0 and v.bruto\_ngtv <>0 then '20' else v.cod\_tipo\_transac end, vr.descrip,v.bruto\_pstv, v.bruto\_ngtv

)

SELECT inter.codigo,

inter.localidad,

inter.fecha,

inter.cod\_tipo\_transac,

sum(inter.pstv) pstv,

sum(inter.ngtv) ngtv,

sum(inter.valormp) valormp,

sum(inter.nro) nro,

sum(inter.cantidadPositiva) cantidadPositiva,

sum(inter.cantidadNegativa) cantidadNegativa

FROM ( SELECT ve.codigo,

ve.localidad,

ve.fecha,

ve.cod\_tipo\_transac,

ve.pstv,

ve.ngtv,

ve.valormp,

COUNT(DISTINCT ve.id\_venta) nro,

SUM(case when va.cantidad > 0 then va.cantidad else 0 end) cantidadPositiva,

SUM(case when va.cantidad < 0 then va.cantidad else 0 end) cantidadNegativa

FROM ve

LEFT JOIN ventas\_articulo va on va.id\_venta = ve.id\_venta

GROUP BY ve.id\_venta, ve.codigo, ve.localidad, ve.fecha, ve.cod\_tipo\_transac, ve.pstv, ve.ngtv, ve.valormp) as inter

GROUP BY inter.codigo, inter.localidad, inter.fecha, inter.cod\_tipo\_transac

END

## Proc\_VentaPorHora

CREATE PROCEDURE [dbo].[Proc\_VentaPorHora]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

convert(varchar(5),datepart(hour,va.date\_created)) + ':00' as Hora,

DATEPART(HOUR, va.date\_created) as HoraSort,

sum(va.cantidad) as NroArticulos,

count(v.id\_venta) as NroTransacciones,

convert(date,va.date\_created) FechaVenta,

SUM(va.valor\_venta) / count(v.id\_venta) as Precio\_Articulo,

SUM(va.cantidad) / count(v.id\_venta) as Transaccion\_Articulo,

sum(va.valor\_venta) as VentaNeta

FROM dbo.ventas\_articulo va

INNER JOIN venta v ON v.id\_venta=va.id\_venta

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(va.cod\_terminal,1,3)

WHERE va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP by l.cod\_localidad, l.descrip, convert(date,va.date\_created), DATEPART(HOUR,va.date\_created)

order by l.cod\_localidad, convert(date,va.date\_created), DATEPART(HOUR,va.date\_created)

END

## Proc\_VentaPorArticulo

CREATE PROCEDURE [dbo].[Proc\_VentaPorArticulo]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

WITH #g AS

(

SELECT ng.descrip nivel, g.id\_grupo, gp.descrip

FROM grupo g

INNER JOIN grupo gp ON g.cod\_grupo = gp.cod\_grupo AND gp.subgrupo1 = '00' AND gp.subgrupo2 = '00' AND gp.subgrupo3 = '00' AND gp.subgrupo4 = '00'

INNER JOIN dbo.grupo\_nivel ng ON ng.id\_grupo\_nivel = gp.grp\_nivel\_id

)

--detalle ventas no efectivas

SELECT c.descrip as ciudad,

l.cod\_localidad as codigo,

l.descrip as localidad,

CONVERT(DATE,v.date\_created) fecha,

#g.nivel nivel,

#g.descrip AS grupo,

a.cod\_imp Cod\_art,

a.descrip1 Descr\_art,

SUM(va.cantidad) AS cantidad,

SUM(va.valor\_venta) AS valor,

va.pctj\_impuesto AS pctj\_impuesto,

SUM(va.impuesto) AS impuesto

FROM venta AS v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

LEFT JOIN dbo.ciudad c ON c.id\_ciudad = l.id\_ciudad

INNER JOIN ventas\_articulo AS va ON va.id\_venta = v.id\_venta

INNER JOIN articulo a ON va.id\_articulo = a.id\_articulo

LEFT JOIN #g ON a.id\_grupo = #g.id\_grupo

WHERE (v.id\_tipo\_transac = '0')

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND bruto\_pstv <> 0

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY c.descrip, l.cod\_localidad, l.descrip, CONVERT(DATE,v.date\_created), #g.nivel, #g.descrip, a.cod\_imp, a.descrip1, va.pctj\_impuesto

HAVING SUM(va.valor\_venta) > 0

ORDER BY fecha, codigo, cod\_art

END

## Proc\_DevolucionPorArticulo

CREATE PROCEDURE [dbo].[Proc\_DevolucionPorArticulo]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

--devoluciones

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal,

CONVERT(date,v.date\_created) Fecha,

v.id\_usuario,

u.nombre + ' ' + u.apellido AS usuario,

a.cod\_imp Cod\_art,

a.descrip1 Descr\_art,

SUM(va.cantidad) AS cantidad,

SUM(va.valor\_venta) AS valor,

va.pctj\_impuesto AS pctj\_impuesto,

SUM(va.impuesto) AS impuesto

FROM venta v

INNER JOIN ventas\_articulo AS va on va.id\_venta = v.id\_venta

LEFT JOIN articulo a ON va.id\_articulo = a.id\_articulo

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

INNER JOIN usuario u ON u.id\_usuario = v.id\_usuario

WHERE ((v.id\_tipo\_transac = '0' AND bruto\_pstv = 0 AND bruto\_ngtv > 0) OR v.id\_tipo\_transac = '20')

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY l.cod\_localidad, l.descrip, v.cod\_terminal, CONVERT(DATE,v.date\_created), v.id\_usuario, u.nombre + ' ' + u.apellido, a.cod\_imp, a.descrip1,va.pctj\_impuesto

ORDER BY l.cod\_localidad, v.cod\_terminal, a.cod\_imp

END

## Proc\_VentaDiaria

CREATE PROCEDURE [dbo].[Proc\_VentaDiaria]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

--detalle ventas efectivas (menos devoluciones)

SELECT c.descrip as ciudad,

l.cod\_localidad as codigo,

l.descrip as localidad,

l.fecha\_apertura,

convert(date,v.date\_created) as fecha,

SUM(v.bruto\_pstv-v.bruto\_ngtv) as valor,

COUNT(v.id\_venta) AS cantidad

FROM venta AS v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

LEFT JOIN ciudad c ON c.id\_ciudad = l.id\_ciudad

WHERE (v.id\_tipo\_transac = '0' OR v.id\_tipo\_transac = '20')

AND (v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal))

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY c.descrip, l.cod\_localidad, l.descrip, l.fecha\_apertura, convert(date,v.date\_created)

END

## Proc\_ImpuestoDiario

CREATE PROCEDURE [dbo].[Proc\_ImpuestoDiario]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT c.descrip as ciudad,

l.cod\_localidad as codigo,

l.descrip as localidad,

l.fecha\_apertura,

convert(date,va.date\_created) as fecha,

SUM(va.impuesto) as impuesto,

COUNT(DISTINCT id\_venta) AS cantidad

FROM ventas\_articulo AS va

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(va.cod\_terminal,1,3)

LEFT JOIN ciudad c ON c.id\_ciudad = l.id\_ciudad

WHERE va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY c.descrip, l.cod\_localidad, l.descrip, l.fecha\_apertura, convert(date,va.date\_created)

END

## Proc\_VentaAnulada

CREATE PROCEDURE [dbo].[Proc\_VentaAnulada]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

--devoluciones

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal,

CONVERT(varchar(19),v.date\_created,20) Fecha,

v.id\_usuario,

u.nombre + ' ' + u.apellido AS usuario,

v.bruto\_pstv AS TotalVenta

FROM venta v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

INNER JOIN usuario u ON u.id\_usuario = v.id\_usuario

where v.id\_tipo\_transac = '7'

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

ORDER BY l.cod\_localidad,v.cod\_terminal, CONVERT(varchar(19),v.date\_created,20) desc

END

## Proc\_ParticipacionArticulo

CREATE PROCEDURE [dbo].[Proc\_ParticipacionArticulo]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT a.cod\_imp Cod\_art,

a.descrip1 Descr\_art,

a.unid\_med\_fleje as UnidadMedida,

SUM(va.cantidad) AS cantidad,

sum(va.valor\_venta) as TotalVenta,

(sum(va.valor\_venta)/t.totalArticulosVendidos\*100) AS PorcentajeParticipacion

FROM venta AS v

INNER JOIN ventas\_articulo va ON va.id\_venta = v.id\_venta

INNER JOIN articulo a ON va.id\_articulo = a.id\_articulo

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

,(select SUM(va.valor\_venta) as TotalArticulosVendidos

from ventas\_articulo va

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(va.cod\_terminal,1,3)

where va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')) AS t

WHERE (v.id\_tipo\_transac = '0' or v.id\_tipo\_transac = '20')

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY a.cod\_imp, a.descrip1, a.unid\_med\_fleje,t.totalArticulosVendidos

HAVING SUM(va.valor\_venta) > 0

ORDER BY SUM(va.valor\_venta) DESC

END

## Proc\_CierreCaja

CREATE PROCEDURE [dbo].[Proc\_CierreCaja]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

WITH inter AS (

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal,

CONVERT(date, v.date\_created) AS fecha,

"cod\_tipo\_transac"=

--cambiar las ventas 0 con valor negativo por devoluciones

CASE

WHEN v.cod\_tipo\_transac = '0' AND v.bruto\_pstv = 0 AND v.bruto\_ngtv <> 0 THEN '20'

ELSE v.cod\_tipo\_transac

END,

vr.descrip,

"pstv" =

CASE

WHEN v.cod\_tipo\_transac = '3' THEN isnull(SUM(vmp.valor), 0)

ELSE v.bruto\_pstv

END,

"ngtv" =

CASE

WHEN v.cod\_tipo\_transac = '4' THEN isnull(SUM(vmp.valor), 0)

ELSE v.bruto\_ngtv

END,

"valormp" =

CASE

WHEN v.cod\_tipo\_transac = '0' THEN isnull(SUM(vmp.valor), 0)

ELSE 0

END,

v.id\_venta,

1 AS nro

FROM venta AS v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

LEFT JOIN ventas\_recogida AS vr ON vr.id\_venta = v.id\_venta

--tipo 3 y 4 en eva1 no tenían info en pstv y negtv sino solo en medios pago

--solo obtener el pago con tarjeta (excluir el efectivo)

LEFT JOIN ventas\_medios\_pago AS vmp ON vmp.id\_venta = v.id\_venta AND (v.id\_tipo\_transac IN ('3', '4') OR vmp.id\_medio\_pago <> '1')

WHERE v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY l.cod\_localidad, l.descrip, v.id\_venta, v.cod\_terminal, CONVERT(date, v.date\_created), v.cod\_tipo\_transac, CASE WHEN v.cod\_tipo\_transac = '0' AND v.bruto\_pstv = 0 AND v.bruto\_ngtv <> 0 THEN '20' ELSE v.cod\_tipo\_transac END, vr.descrip, v.bruto\_pstv, v.bruto\_ngtv

)

SELECT codigo,

localidad,

cod\_terminal,

fecha,

cod\_tipo\_transac,

descrip,

SUM(pstv) pstv,

SUM(ngtv) ngtv,

SUM(valormp) valormp,

COUNT (DISTINCT id\_venta) AS nro

from inter

GROUP BY codigo, localidad, cod\_terminal, fecha, cod\_tipo\_transac, descrip

END

## Proc\_VentaPorUsuario

CREATE PROCEDURE Proc\_VentaPorUsuario

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal,

v.id\_usuario,

u.nombre + ' ' + u.apellido AS usuario,

CONVERT(date, v.date\_created) AS fecha,

"cod\_tipo\_transac"=

--cambiar las ventas 0 con valor negativo por devoluciones

CASE

WHEN v.cod\_tipo\_transac = '0' AND v.bruto\_pstv = 0 AND v.bruto\_ngtv <> 0 THEN '20'

ELSE v.cod\_tipo\_transac

END,

vr.descrip,

"pstv" =

CASE

WHEN v.cod\_tipo\_transac = '3' THEN isnull(SUM(vmp.valor), 0)

ELSE v.bruto\_pstv

END,

"ngtv" =

CASE

WHEN v.cod\_tipo\_transac = '4' THEN isnull(SUM(vmp.valor), 0)

ELSE v.bruto\_ngtv

END,

"valormp" =

CASE

WHEN v.cod\_tipo\_transac = '0' THEN isnull(SUM(vmp.valor), 0)

ELSE 0

END,

1 AS nro

FROM venta AS v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

INNER JOIN usuario u ON u.id\_usuario = v.id\_usuario

LEFT JOIN ventas\_recogida AS vr ON vr.id\_venta = v.id\_venta

--tipo 3 y 4 en eva1 no tenían info en pstv y negtv sino solo en medios pago

--solo obtener el pago con tarjeta (excluir el efectivo)

LEFT JOIN ventas\_medios\_pago AS vmp ON vmp.id\_venta = v.id\_venta AND (v.id\_tipo\_transac IN ('3', '4') OR vmp.id\_medio\_pago <> '1')

WHERE v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY l.cod\_localidad, l.descrip, v.id\_usuario, u.nombre + ' ' + u.apellido, v.id\_venta, v.cod\_terminal, CONVERT(date, v.date\_created), v.cod\_tipo\_transac, CASE WHEN v.cod\_tipo\_transac = '0' AND v.bruto\_pstv = 0 AND v.bruto\_ngtv <> 0 THEN '20' ELSE v.cod\_tipo\_transac END, vr.descrip, v.bruto\_pstv, v.bruto\_ngtv

END

## Proc\_VentasCliente

CREATE PROCEDURE [dbo].[Proc\_VentasCliente]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal,

CONVERT(date,v.date\_created) Fecha,

v.id\_usuario,

u.nombre + ' ' + u.apellido AS usuario,

v.prefijo as prefijo,

v.nro\_fact as nro\_factura,

p.id\_persona as id\_cliente,

p.primer\_nombre + ' ' + p.primer\_apellido as cliente,

SUM(v.bruto\_pstv - v.bruto\_ngtv) AS valor\_venta

FROM venta v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

INNER JOIN usuario u ON u.id\_usuario = v.id\_usuario

INNER JOIN ventas\_cliente vc ON vc.id\_venta = v.id\_venta

LEFT JOIN persona p ON p.id\_persona = vc.cliente\_id

WHERE v.id\_tipo\_transac = '0'

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND @Localidades LIKE ('%' + l.cod\_localidad + '%')

GROUP BY l.cod\_localidad, l.descrip, v.cod\_terminal, CONVERT(date,v.date\_created), v.id\_usuario, u.nombre + ' ' + u.apellido, v.prefijo, v.nro\_fact, p.id\_persona, p.primer\_nombre + ' ' + p.primer\_apellido

ORDER BY l.cod\_localidad ASC, CONVERT(date,v.date\_created) ASC

END

## Proc\_VentaPorCodBarra

CREATE PROCEDURE [dbo].[Proc\_VentaPorCodBarra]

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date,

@Localidades varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT a.cod\_imp Cod\_art,

a.descrip1 Descr\_art,

va.cod\_barra,

SUM(va.cantidad) AS cantidad,

SUM(va.valor\_venta) as TotalVenta

FROM ventas\_articulo va

LEFT JOIN articulo a ON va.id\_articulo = a.id\_articulo

INNER JOIN venta v ON va.id\_venta = v.id\_venta

WHERE va.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND v.id\_tipo\_transac = '0'

AND @Localidades LIKE ('%' + SUBSTRING(va.cod\_terminal,1,3) + '%')

and a.cod\_imp != va.cod\_barra

END

## Proc\_Exc\_VentasBonos

CREATE PROCEDURE Proc\_Exc\_VentasBonos

-- Add the parameters for the stored procedure here

@FechaInicial date,

@FechaFinal date

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT v.id\_venta,

l.cod\_localidad as codigo,

l.descrip as localidad,

v.cod\_terminal terminal,

v.date\_created Fecha,

v.nro\_transac,

v.nro\_fact,

v.prefijo,

v.id\_usuario,

u.nombre + ' ' + u.apellido AS usuario,

SUM(va.cantidad) AS CantBonos,

v.bruto\_pstv-v.bruto\_ngtv as valor\_venta

FROM venta AS v

INNER JOIN localidad l ON l.cod\_localidad = SUBSTRING(v.cod\_terminal,1,3)

INNER JOIN usuario u ON u.id\_usuario = v.id\_usuario

INNER JOIN ventas\_articulo va ON v.id\_venta = va.id\_venta

WHERE v.id\_tipo\_transac = '0'

AND bruto\_pstv <> 0

AND v.date\_created BETWEEN DATEADD(dd,0,@fechaInicial) AND DATEADD(dd,1,@fechaFinal)

AND va.cod\_imp = '2' --código del artículo correspondiente al bono.

GROUP BY v.id\_venta, l.cod\_localidad, l.descrip, v.cod\_terminal, v.date\_created, v.nro\_transac, v.nro\_fact, v.prefijo, v.id\_usuario, u.nombre + ' ' + u.apellido, v.bruto\_pstv, v.bruto\_ngtv

HAVING SUM(va.cantidad) > 0

AND SUM(va.cantidad) > ((v.bruto\_pstv-v.bruto\_ngtv) + (SUM(va.cantidad) \* 1000)) / 20000 -- 1 bono de $ 1.000 por cada $20.000

END

## Proc\_Empresa

CREATE PROCEDURE Proc\_Empresa

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT nombre FROM empresa

END

## Proc\_Ciudades

CREATE PROCEDURE Proc\_Ciudades

-- Add the parameters for the stored procedure here

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

SELECT id\_ciudad, descrip

FROM ciudad

END

GO

## Proc\_Localidadeses

CREATE PROCEDURE Proc\_Localidadeses

-- Add the parameters for the stored procedure here

@Ciudad varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

select l.cod\_localidad Codigo\_Localidad,

l.cod\_localidad + ' - ' + l.descrip CodLocalidad

from dbo.localidad l

LEFT JOIN ciudad c ON c.id\_ciudad = l.id\_ciudad

where tipo = 'LOCAL' and cerrada = 0 and cod\_localidad <> 'C01'

AND @Ciudad LIKE ('%' + c.id\_ciudad + '%')

END

# Índices

Los siguientes índices mejoran el rendimiento de los reportes:

* Contabilidad consolidada.
* Cierre de caja.
* Impuesto diario.
* Ventas por usuario

## Índice sobre venta

CREATE NONCLUSTERED INDEX [idx\_merca\_fecha\_creacion] ON [dbo].[venta]

(

[date\_created] ASC

)

INCLUDE (

[id\_venta],

[bruto\_ngtv],

[bruto\_pstv],

[cod\_terminal],

[cod\_tipo\_transac],

[id\_tipo\_transac],

[id\_terminal],

[id\_usuario]) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, FILLFACTOR = 80)

GO

## Índice sobre ventas\_articulo

CREATE NONCLUSTERED INDEX [idx\_ventas\_articulo] ON [dbo].[ventas\_articulo]

(

[date\_created] ASC

)

INCLUDE ( [id\_articulo],

[cantidad],

[cod\_terminal],

[impuesto],

[valor\_venta],

[id\_venta]) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, FILLFACTOR = 80)

GO

## Índice sobre ventas\_medios\_pago

CREATE NONCLUSTERED INDEX [idx\_medio\_pago] ON [ventas\_medios\_pago]

(

[id\_venta]

)

INCLUDE (

[id\_medio\_pago],

[valor]) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, FILLFACTOR = 80)

GO

## Índice sobre ventas\_recogida

CREATE NONCLUSTERED INDEX [idx\_descr\_recogida] ON [ventas\_recogida]

(

[id\_venta]

)

INCLUDE (

[descrip]) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, FILLFACTOR = 80)

GO