



UNIVERSIDAD NACIONAL AUTONOMA DE HONDURAS
CIUDAD UNIVERSITARIA JOSE TRINIDAD REYES



Facultad de Ingeniería

Carrera de Ingeniería en Sistemas

Asignatura:
IS501 Bases de Datos I

Catedrática:
Ing. Cindy Euceda

Alumno:
Luis Daniel Mejía valladares

N° Cuenta:
20181006117

Sección:
1800

Unidad:
III

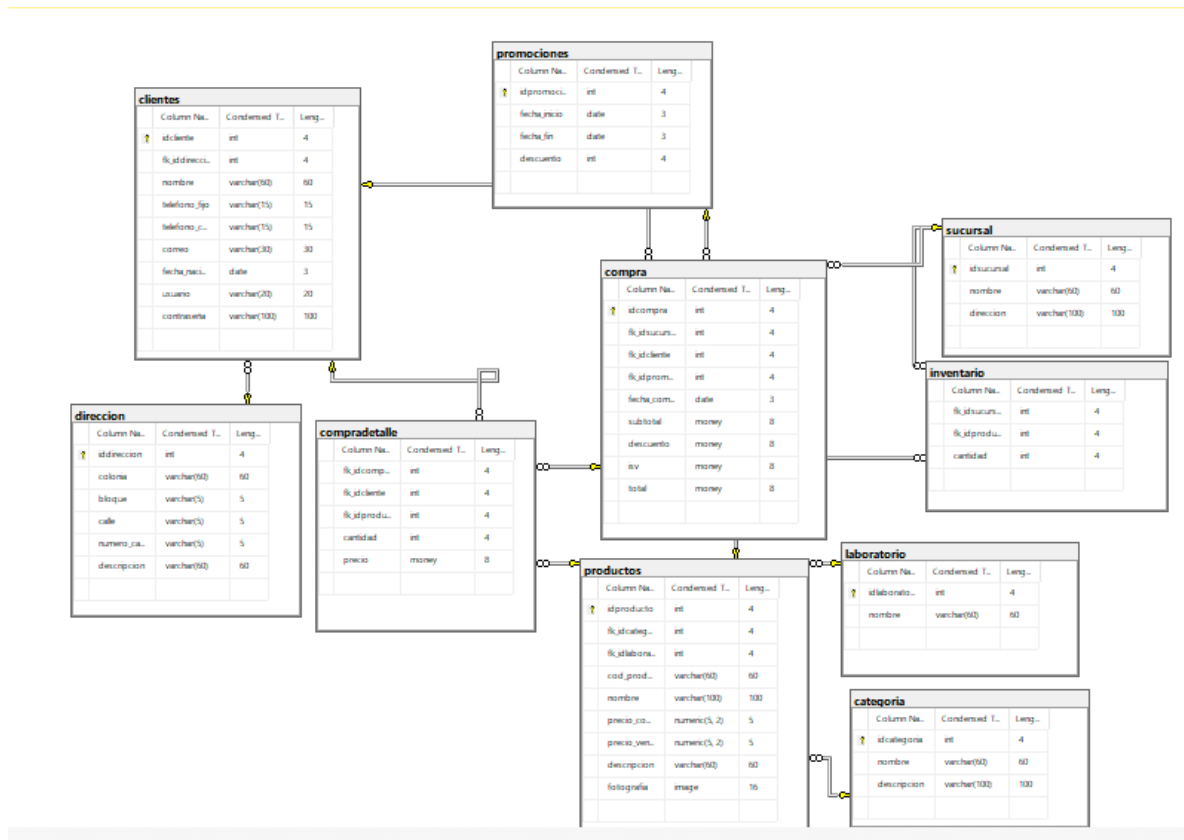
Ciudad Universitaria José Trinidad Reyes, Tegucigalpa MDC, 30 de abril de 2022

INTRODUCCION

En el presente informe de proyecto se muestra la información necesaria para la creación de la base de datos farmacia, conforme a lo enunciado en el ejercicio de proyecto, se da a conocer el diagrama de las tablas de la base de datos, y diferentes reportes generados a través de consultas, vistas y procedimientos almacenados.

DESARROLLO TECNICO

DIAGRAMA RELEACIONAL



SCRIPT DE CREACIÓN DE LA BASE DE DATOS

```
USE [master]
GO

/***** Object: Database [farmacia] *****/
CREATE DATABASE [farmacia]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'farmacia', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\farmacia.mdf' , SIZE = 8192KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'farmacia_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\farmacia_log.ldf' , SIZE = 8192KB , MAXSIZE =
2048GB , FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [farmacia].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
```

```
ALTER DATABASE [farmacia] SET ANSI_NULL_DEFAULT OFF
GO
```

```
ALTER DATABASE [farmacia] SET ANSI_NULLS OFF
GO
```

```
ALTER DATABASE [farmacia] SET ANSI_PADDING OFF
GO
```

```
ALTER DATABASE [farmacia] SET ANSI_WARNINGS OFF
GO
```

```
ALTER DATABASE [farmacia] SET ARITHABORT OFF
GO
```

```
ALTER DATABASE [farmacia] SET AUTO_CLOSE OFF
GO
```

```
ALTER DATABASE [farmacia] SET AUTO_SHRINK OFF
GO
```

```
ALTER DATABASE [farmacia] SET AUTO_UPDATE_STATISTICS ON
GO
```

```
ALTER DATABASE [farmacia] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
```

```
ALTER DATABASE [farmacia] SET CURSOR_DEFAULT GLOBAL
GO
```

```
ALTER DATABASE [farmacia] SET CONCAT_NULL_YIELDS_NULL OFF
GO
```

```
ALTER DATABASE [farmacia] SET NUMERIC_ROUNDABORT OFF
GO
```

```
ALTER DATABASE [farmacia] SET QUOTED_IDENTIFIER OFF
GO
```

```
ALTER DATABASE [farmacia] SET RECURSIVE_TRIGGERS OFF
GO
```

```
ALTER DATABASE [farmacia] SET ENABLE_BROKER
GO
```

```
ALTER DATABASE [farmacia] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
```

```
ALTER DATABASE [farmacia] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
```

```
ALTER DATABASE [farmacia] SET TRUSTWORTHY OFF
GO
```

```
ALTER DATABASE [farmacia] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
```

```

ALTER DATABASE [farmacia] SET PARAMETERIZATION SIMPLE
GO

ALTER DATABASE [farmacia] SET READ_COMMITTED_SNAPSHOT OFF
GO

ALTER DATABASE [farmacia] SET HONOR_BROKER_PRIORITY OFF
GO

ALTER DATABASE [farmacia] SET RECOVERY FULL
GO

ALTER DATABASE [farmacia] SET MULTI_USER
GO

ALTER DATABASE [farmacia] SET PAGE_VERIFY CHECKSUM
GO

ALTER DATABASE [farmacia] SET DB_CHAINING OFF
GO

ALTER DATABASE [farmacia] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO

ALTER DATABASE [farmacia] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO

ALTER DATABASE [farmacia] SET DELAYED_DURABILITY = DISABLED
GO

ALTER DATABASE [farmacia] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO

ALTER DATABASE [farmacia] SET QUERY_STORE = OFF
GO

ALTER DATABASE [farmacia] SET READ_WRITE
GO

USE [farmacia]
GO

/***** Object: Table [dbo].[categoria] *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[categoria](
    [idcategoria] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](60) NOT NULL,
    [descripcion] [varchar](100) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idcategoria] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[direccion] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[direccion](
    [iddireccion] [int] IDENTITY(1,1) NOT NULL,
    [colonia] [varchar](60) NOT NULL,
    [bloque] [varchar](5) NOT NULL,
    [calle] [varchar](5) NOT NULL,
    [numero_casa] [varchar](5) NOT NULL,
    [descripcion] [varchar](60) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [iddireccion] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[clientes] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[clientes](
    [idcliente] [int] IDENTITY(1,1) NOT NULL,
    [fk_iddireccion] [int] NOT NULL,
    [nombre] [varchar](60) NOT NULL,
    [telefono_fijo] [varchar](15) NULL,
    [telefono_celular] [varchar](15) NULL,
    [correo] [varchar](30) NOT NULL,
    [fecha_nacimiento] [date] NOT NULL,
    [usuario] [varchar](20) NOT NULL,
    [contraseña] [varchar](100) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idcliente] ASC
)
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[clientes] WITH CHECK ADD FOREIGN KEY([fk_iddireccion])
REFERENCES [dbo].[direccion] ([iddireccion])
GO
```

```
USE [farmacia]
GO
```

```
/***** Object: Table [dbo].[compra] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[compra](
    [idcompra] [int] IDENTITY(1,1) NOT NULL,
    [fk_idsucursal] [int] NOT NULL,
    [fk_idcliente] [int] NOT NULL,
    [fk_idpromocion] [int] NOT NULL,
    [fecha_compra] [date] NOT NULL,
    [subtotal] [money] NOT NULL,
    [descuento] [money] NOT NULL,
    [isv] [money] NOT NULL,
    [total] [money] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idcompra] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[compra] WITH CHECK ADD FOREIGN KEY([fk_idcliente])
REFERENCES [dbo].[clientes] ([idcliente])
GO
```

```
ALTER TABLE [dbo].[compra] WITH CHECK ADD FOREIGN KEY([fk_idpromocion])
REFERENCES [dbo].[promociones] ([idpromocion])
GO
```

```
ALTER TABLE [dbo].[compra] WITH CHECK ADD FOREIGN KEY([fk_idsucursal])
REFERENCES [dbo].[sucursal] ([idsucursal])
GO
```

```
USE [farmacia]
GO
```

```
/***** Object: Table [dbo].[compradetalle] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[compradetalle](
    [fk_idcompra] [int] NOT NULL,
    [fk_idcliente] [int] NOT NULL,
    [fk_idproducto] [int] NOT NULL,
    [cantidad] [int] NOT NULL,
    [precio] [money] NOT NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[compradetalle] WITH CHECK ADD FOREIGN KEY([fk_idcompra])
REFERENCES [dbo].[compra] ([idcompra])
GO
```

```
ALTER TABLE [dbo].[compradetalle] WITH CHECK ADD FOREIGN KEY([fk_idcliente])
REFERENCES [dbo].[clientes] ([idcliente])
GO
```

```
ALTER TABLE [dbo].[compradetalle] WITH CHECK ADD FOREIGN KEY([fk_idproducto])
REFERENCES [dbo].[productos] ([idproducto])
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[sucursal] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[sucursal](
    [idsucursal] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](60) NOT NULL,
    [direccion] [varchar](100) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idsucursal] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[inventario] *****/
SET ANSI_NULLS ON
GO
```



```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[inventario](
    [fk_idsucursal] [int] NOT NULL,
    [fk_idproducto] [int] NOT NULL,
    [cantidad] [int] NOT NULL
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[inventario] WITH CHECK ADD FOREIGN KEY([fk_idsucursal])
REFERENCES [dbo].[sucursal] ([idsucursal])
GO
```

```
ALTER TABLE [dbo].[inventario] WITH CHECK ADD FOREIGN KEY([fk_idproducto])
REFERENCES [dbo].[productos] ([idproducto])
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[laboratorio] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[laboratorio](
    [idlaboratorio] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](60) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idlaboratorio] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```
USE [farmacia]
GO
```

```
/****** Object: Table [dbo].[productos] *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[productos](
    [idproducto] [int] IDENTITY(1,1) NOT NULL,
    [fk_idcategoria] [int] NOT NULL,
    [fk_idlaboratorio] [int] NOT NULL,
    [cod_producto] [varchar](60) NOT NULL,
    [nombre] [varchar](100) NOT NULL,
    [precio_compra] [numeric](5, 2) NOT NULL,
```

```

        [precio_venta] [numeric](5, 2) NOT NULL,
        [descripcion] [varchar](60) NOT NULL,
        [fotografia] [image] NULL,
PRIMARY KEY CLUSTERED
(
    [idproducto] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[productos] WITH CHECK ADD FOREIGN KEY([fk_idcategoria])
REFERENCES [dbo].[categoria] ([idcategoria])
GO

ALTER TABLE [dbo].[productos] WITH CHECK ADD FOREIGN KEY([fk_idlaboratorio])
REFERENCES [dbo].[laboratorio] ([idlaboratorio])
GO

USE [farmacia]
GO

/***** Object: Table [dbo].[promociones] *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[promociones](
    [idpromocion] [int] IDENTITY(1,1) NOT NULL,
    [fecha_inicio] [date] NOT NULL,
    [fecha_fin] [date] NOT NULL,
    [descuento] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [idpromocion] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

```

REPORTES

--Consultas

--i. Edad promedio de los clientes

```
select avg(DATEDIFF(YEAR, fecha_nacimiento, GETDATE())) as Promedio_Edad from clientes
```

-- ii. 10 productos más vendidos

```
select top 10 fk_idproducto, sum(cantidad) Cantidad from compradetalle group by  
fk_idproducto order by sum(cantidad) desc
```

-- iii. Cantidad de productos por categoría

```
select cat.nombre, sum(cantidad) Cantidad from inventario iv  
inner join productos pr on iv.fk_idproducto=pr.idproducto  
inner join categoria cat on cat.idcategoria=pr.fk_idcategoria  
group by cat.nombre
```

-- iv. Listar los productos más bajos en ventas

```
select top 10 fk_idproducto, sum(cantidad) Cantidad from compradetalle group by  
fk_idproducto order by sum(cantidad) asc
```

-- v. Listado de productos agrupados por categoría

```
select cat.nombre, count(*) Cantidad from productos pr  
inner join categoria cat on cat.idcategoria=pr.fk_idcategoria  
group by cat.nombre
```

--Vistas

-- i. Reporte mensual de ventas por sucursal

```
create view VW_Ventas_Sucursal  
as  
select s.nombre, sum(total) total_ventas from compra c  
inner join sucursal s on s.idsucursal=c.fk_idsucursal  
group by s.nombre
```

```
select * from VW_Ventas_Sucursal
```

-- ii. Reporte de ventas mensual por producto

```
create view VW_Ventas_Producto  
as  
select p.nombre, sum(cantidad) total_ventas from compra c  
inner join compradetalle cd on cd.fk_idcliente=c.fk_idcliente and  
cd.fk_idcompra=c.idcompra  
inner join productos p on p.idproducto=cd.fk_idproducto  
group by p.nombre
```

```
select * from VW_Ventas_Producto
```

-- iii. Total ventas por categoría

```
create view VW_Ventas_Categoria  
as  
select cat.nombre, sum(cantidad) total_ventas from compra c  
inner join compradetalle cd on cd.fk_idcliente=c.fk_idcliente and  
cd.fk_idcompra=c.idcompra  
inner join productos p on p.idproducto=cd.fk_idproducto  
inner join categoria cat on p.fk_idcategoria=cat.idcategoria  
group by cat.nombre
```

```
select * from VW_Ventas_Categoria
```

```

-- iv. Productos en el inventario con cantidad menor que 10 unidades por sucursal
create view VW_Inventario_Menor_10
as
select s.nombre Sucursal, p.nombre Producto, sum(inv.cantidad)Cantidad from
inventario inv
inner join productos p on p.idproducto=inv.fk_idproducto
inner join sucursal s on s.idsucursal=inv.fk_idsucursal
group by s.nombre, p.nombre
having sum(inv.cantidad)<10

select * from VW_Inventario_Menor_10

-- v. Listado de clientes que no han realizado ningún pedido durante el mes
create view VM_Clientes_Sin_Pedido as
select nombre, telefono_fijo, telefono_celular, correo from clientes
where idcliente not in (select fk_idcliente from compra where
month(fecha_compra)=MONTH(GETDATE()))

select * from VM_Clientes_Sin_Pedido

--c. Procedimientos almacenados
-- i. Función para búsqueda de clientes por nombre

create procedure SP_Buscar_Cliente (@nombre varchar(10))
as
begin
select nombre, telefono_fijo, telefono_celular, correo from clientes where nombre
like '%' + @nombre + '%'
end

execute SP_Buscar_Cliente 'Luis'

-- ii. Eliminar clientes por id
create procedure SP_Eliminar_Cliente (@idcliente int)
as
begin
delete from clientes where idcliente=@idcliente
end

execute SP_Eliminar_Cliente 5

-- iii. Actualizar clientes por id
create procedure SP_Actualizar_Cliente
(@idcliente int, @fk_iddireccion int, @nombre varchar(60), @telefono_fijo
varchar(15),
@telefono_celular varchar(15), @correo varchar(30), @fecha_nacimiento date,
@usuario varchar(20), @contraseña varchar(100) )
as
begin
update clientes set
fk_iddireccion=@fk_iddireccion,
nombre=@nombre,
telefono_fijo=@telefono_fijo,
telefono_celular=@telefono_celular,
correo=@correo,
fecha_nacimiento=@fecha_nacimiento,
usuario=@usuario,
contraseña=@contraseña

```

```
where idcliente=@idcliente  
end
```

```
execute SP_Actualizar_Cliente 4, 3, 'Prueba', 'Prueba', 'Prueba', 'Prueba',  
'29/02/1988', 'Prueba', 'Prueba'
```

```
-- iv. Listado de todos los pedidos realizados por un cliente
```

```
create procedure SP_Buscar_Pedidos_Cliente (@idcliente int)  
as  
begin  
select c.* from clientes cl  
inner join compra c on cl.idcliente=c.fk_idcliente  
where cl.idcliente=@idcliente  
end
```

```
execute SP_Buscar_Pedidos_Cliente 1
```

```
-- v. Total de ventas de un producto en una fecha determinada
```

```
create procedure SP_Ventas_Producto (@fechaInicio date, @fechaFin date, @idproducto  
int)  
as  
begin  
select p.nombre, sum(cd.precio) total_ventas from compra c  
inner join compradetalle cd on cd.fk_idcompra=c.idcompra  
inner join productos p on p.idproducto=c.fk_idsucursal  
where c.fecha_compra between @fechaInicio and @fechaFin and p.idproducto=@idproducto  
group by p.nombre  
end
```

```
execute SP_Ventas_Producto '01/04/2022', '30/04/2022', 1
```