```
-----------------------------------------------------
--      FUNCIONES ESCALARES
-- -------------------------------------------------

--1)Crear una función que devuelva el promedio de los productos.
--Tablas: Production.Product
--Campos: ListPrice
GO
CREATE FUNCTION dbo.FN_Promedio()
RETURNS MONEY
AS
BEGIN
              DECLARE @promedio MONEY;

              SELECT
                            @promedio=AVG(ListPrice)
              FROM
                            Production.Product

              RETURN @promedio
END
GO

SELECT dbo.FN_Promedio()

--2)Crear una función que dado un código de producto devuelva el total de ventas
para dicho producto luego,
--mediante una  consulta, traer código y total de ventas.
--Tablas: Sales.SalesOrderDetail
--Campos: ProductID, LineTotal

GO
CREATE FUNCTION dbo.FN_VentasProductos(@codigoProducto INT)
RETURNS INT
AS
 BEGIN
          DECLARE @total INT;

          SELECT
                      @total = SUM(LineTotal)
          FROM
                      Sales.SalesOrderDetail
          WHERE
                      ProductID= @codigoProducto

          IF (@total IS NULL)
                SET @total = 0

          RETURN @total
 END
GO

SELECT ProductID, dbo.FN_VentasProductos(712)
FROM Sales.SalesOrderDetail
WHERE ProductID=712
GO
```

```
--3) Crear una función que dado un código devuelva la cantidad de productos
vendidos o cero si no se ha vendido.
--Tablas: Sales.SalesOrderDetail
--Campos: ProductID, OrderQty

CREATE FUNCTION dbo.FN_CantidadVentasProductos(@codigoProducto INT)
RETURNS INT
AS
 BEGIN
          DECLARE @total INT;

          SELECT
                    @total = SUM(OrderQty)
          FROM
                    Sales.SalesOrderDetail
          WHERE
                    ProductID= @codigoProducto

          IF (@total IS NULL)
                SET @total = 0

          RETURN @total
 END
GO

SELECT dbo.FN_CantidadVentasProductos(712)



--4)Crear una función que devuelva el promedio de venta
-- luego obtner los productos cuyo precio sea inferior al promedio.
--Tablas: Sales.SalesOrderDetail, Production.Product
--Campos: ProductID, ListPrice
GO
CREATE FUNCTION dbo.FN_TotalVentasProductos()
RETURNS INT
AS
 BEGIN
          DECLARE @total INT;

          SELECT
                    @total = avg(LineTotal)
          FROM
                    Sales.SalesOrderDetail

          RETURN @total
 END
GO

SELECT ProductID
FROM Production.Product
WHERE ListPrice<dbo.FN_TotalVentasProductos()
```

```
-------------------------------------------------------------------------------
------
--      FUNCIONES DE TABLA EN LÍNEA
-- -----------------------------------------------------------------------------
------
--5) Crear una función que dado un año, devuelva nombre y  apellido de los
empleados
--que ingresaron ese año
--Tablas: Person.Person, HumanResources.Employee
--Campos: FirstName, LastName,HireDate, BusinessEntityID
GO
CREATE FUNCTION  dbo.IF_IngresoEmpleadosAnuales(@año INT)
RETURNS TABLE
AS
        RETURN
        (
                SELECT
                                FirstName
                                ,LastName
                                ,HireDate
                FROM
                                Person.Person p

                                INNER JOIN HumanResources.Employee e
                                ON e.BusinessEntityID= p.BusinessEntityID
                WHERE
                                YEAR(HireDate)=@año
        )
GO

SELECT *
FROM dbo.IF_IngresoEmpleadosAnuales(2011)



--6) Crear una función que reciba un parámetro correspondiente a un precio y nos
retorna
--una tabla con  código,nombre, color y precio de todos los productos cuyo precio
sea inferior al parámetro  ingresado
--Tablas: Production.Product
--Campos: ProductID, Name, Color, ListPrice
GO
CREATE FUNCTION dbo.if_PrecioProducto(@precio money)
RETURNS TABLE
AS
RETURN
(
    SELECT
                        ProductID
                        ,Name
                        ,Color
                        ,ListPrice
    FROM
                        Production.Product AS P
    WHERE
                        ListPrice<@precio
)
GO

SELECT          *
FROM            dbo.if_PrecioProducto (1340)
ORDER BY        ProductID, Name   DESC;
```

```
-------------------------------------------------------------------------------------
------
--      FUNCIONES DE MULTI SENTENCIA
-- -----------------------------------------------------------------------------------
------
--7)Realizar el mismo pedido que en el punto anterior pero utilizando este  tipo de
función
--Tablas: Production.Product
--Campos: ProductID, Name, Color, ListPrice
GO
CREATE FUNCTION dbo.tf_ofertas( @minimo DECIMAL(6,2))
RETURNS @oferta
TABLE
(
       Codigo         INT
       ,Nombre        VARCHAR(40)
       ,Color         VARCHAR(30)
       ,Precio        DECIMAL(6,2)
)
AS
BEGIN
       INSERT @oferta
       SELECT ProductID
                     ,Name
                     ,Color
                     ,ListPrice
       FROM
                     Production.Product
```