



Nombre:

05/08/2021

BRUNO BELTRE GUZMAN

Curso:

Introducción a las Bases de Datos

Profesora:

Naïomi Solano

Trabajo Final

Proyecto final introducción a las bases de datos

Escoger un negocio comercial que se dedique a la venta de productos, debe escoger una temática por ejemplo ferretería, farmacia, tienda deportiva o de ropa, etc. después de haber escogido su tema realice las siguientes actividades.

El proyecto final consiste en crear una base de datos para el negocio escogido por el estudiante. Debe entregar.

1) Entregar documentación con objetivos, breve descripción del negocio.

TecnoDigital

TecnoDigital es una miniempresa dominicana dedicada a prestar servicios en la venta de artículos tecnológicos y soporte de dispositivos tecnológicos. La empresa presta servicios de venta de artículos tecnológicos de forma virtual.

El origen de esta empresa surge en el presente año 2021 iniciando como una pequeña empresa de venta de accesorios para computadores, accesorios para dispositivos móviles y equipos de oficinas con la gran meta de ser una gran empresa.

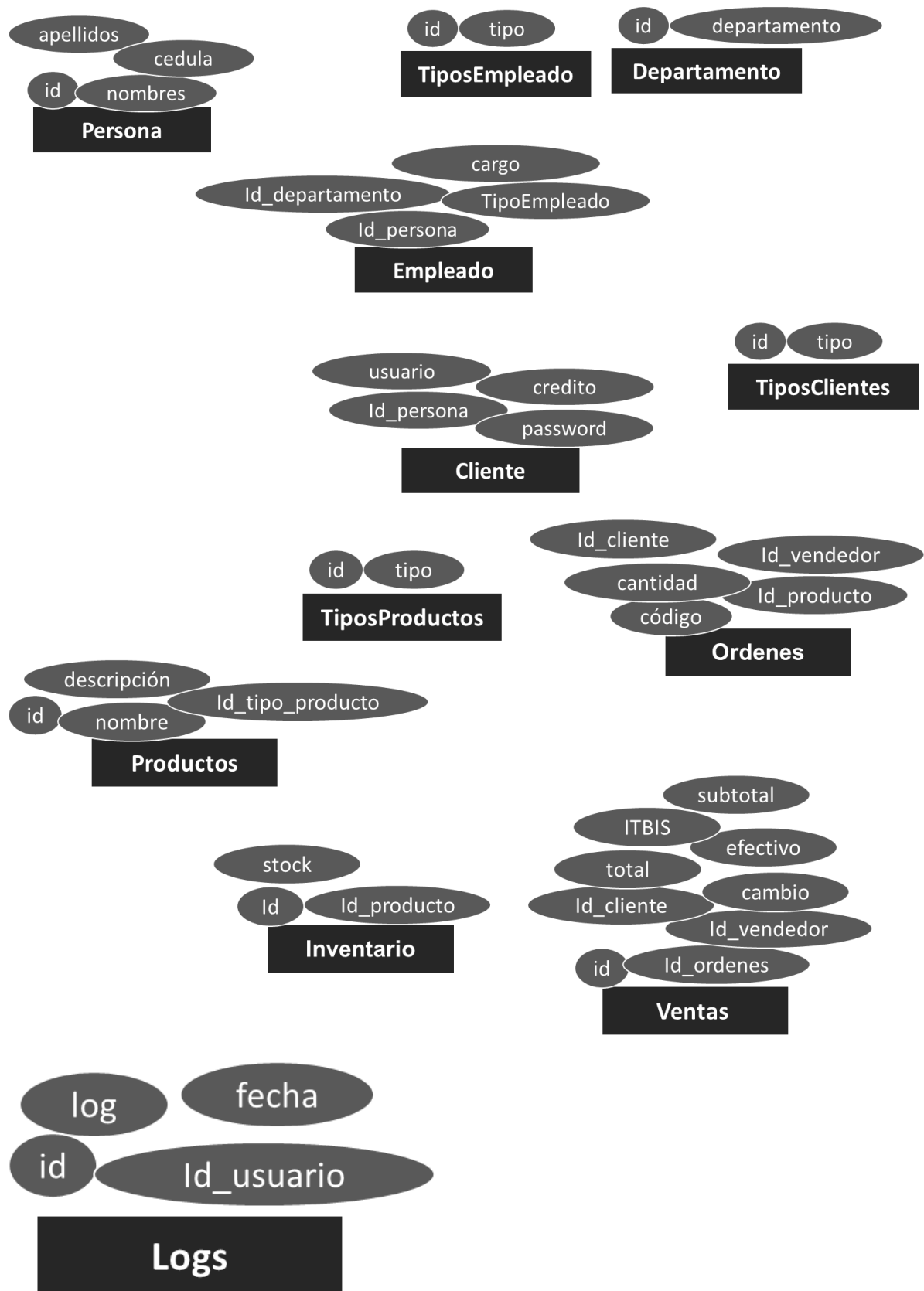
Objetivos

Nuestra empresa fue creada con el objetivo vender artículos tecnológicos que pueden satisfacer las necesidades de los clientes.

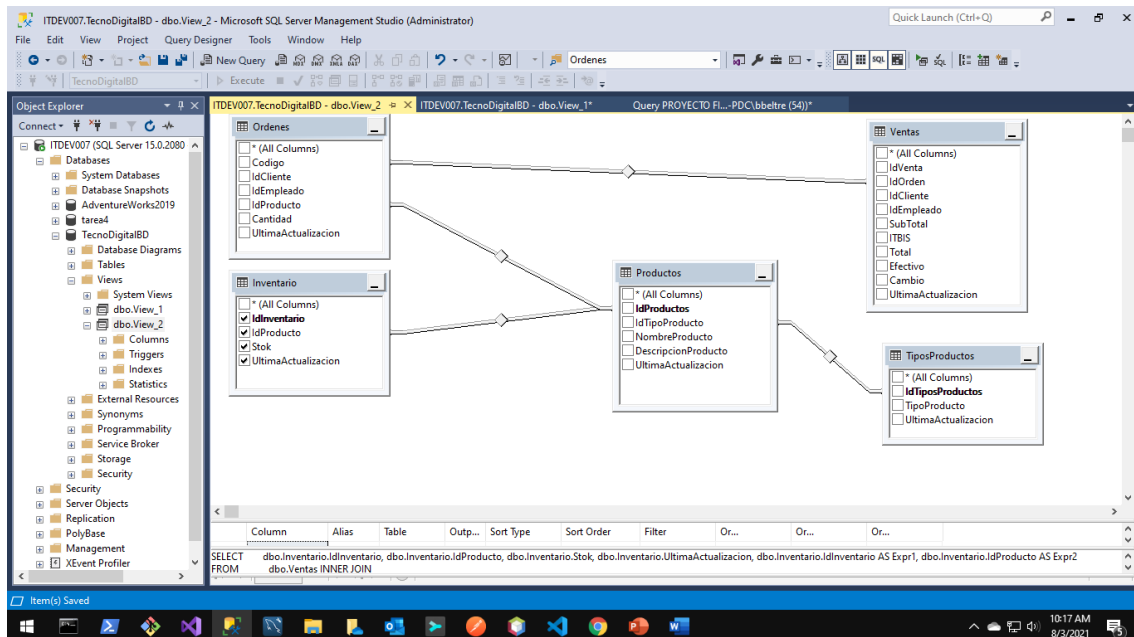
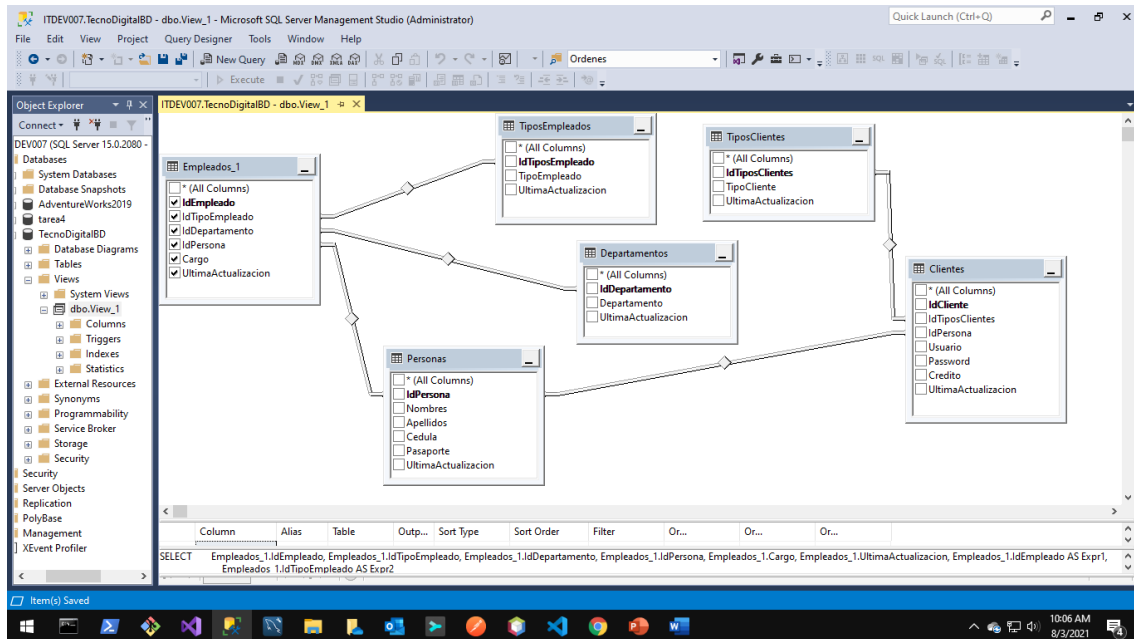
Otro de nuestros objetivos es poder brindar un servicio optimizado, de tal manera que el cliente pueda adquirir su producto de una forma fácil y sencilla.

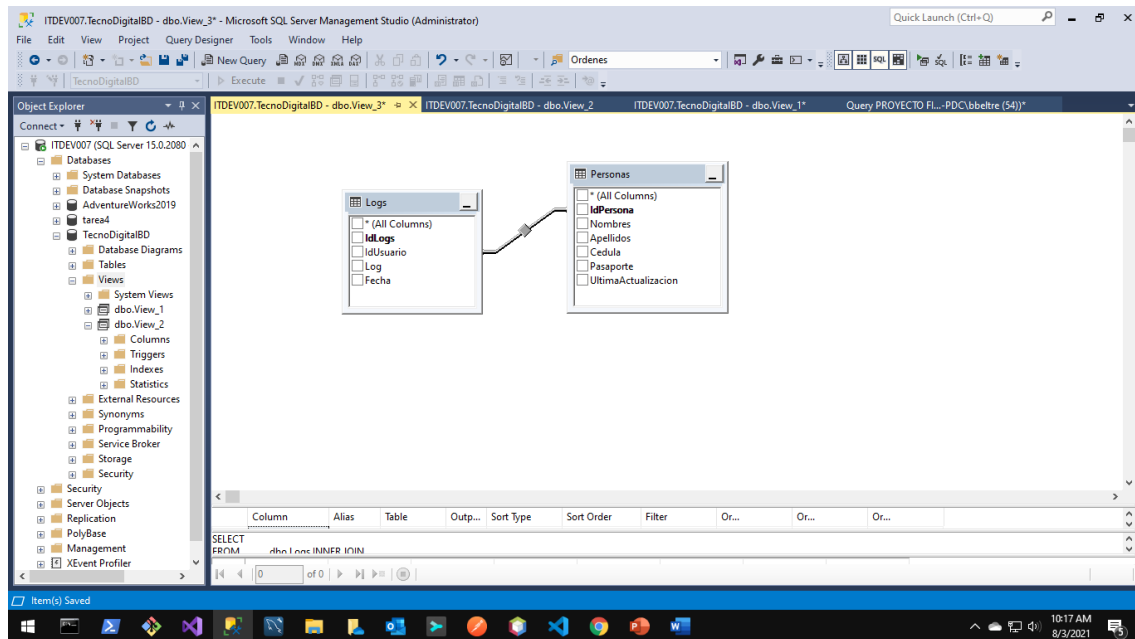
Brindar un servicio más rápido al cliente y mejorar en todos los ámbitos a la hora de realizar una compra para que sea objetiva.

2) Realizar diagrama de entidad relación



3) Diagrama modelo relacional. (realizado en SQL server managment studio)





4) Query de creación de bases de datos, tablas, relaciones, vistas, procedimientos almacenados.

/*

4) Query de creación de bases de datos, tablas, relaciones, vistas, procedimientos almacenados.

*/

--create database TecnoDigitalBD;

use TecnoDigitalBD;

/*

Productos

*/

DROP TABLE if EXISTS Productos;

CREATE TABLE Productos (

IdProducto int Identity(1, 1) primary key not null,

IdTipoProducto int not null,

NombreProducto varchar(50) not null ,

DescripcionProducto varchar(50) default null,

Precio float not null,

UltimaActualizacion datetime not null default GETDATE())

);

/*

TiposProductos

*/

DROP TABLE if EXISTS TiposProductos;

CREATE TABLE TiposProductos (

IdTiposProductos int Identity(1, 1) primary key not null,

TipoProducto varchar(50) not null ,

UltimaActualizacion datetime not null default GETDATE()

);

/*

Persona

*/

DROP TABLE if EXISTS Personas;

CREATE TABLE Personas (

IdPersona int Identity(1, 1) primary key not null,

Nombres varchar(50) not null,

Apellidos varchar(50) not null,

Cedula varchar(50) not null ,

Pasaporte varchar(50) default null,

UltimaActualizacion datetime not null default GETDATE()

);

/*

Empleado

*/

DROP TABLE if EXISTS Empleados;

CREATE TABLE Empleados (

IdPersona int not null,

IdTipoEmpleado int not null,

```
        IdDepartamento int not null,  
        Cargo varchar(50) not null,  
        Usuario varchar(50) not null ,  
        Password varchar(255) not null,  
        UltimaActualizacion datetime not null default GETDATE()  
    );
```

```
/*
```

```
    TiposEmpleado
```

```
*/
```

```
DROP TABLE if EXISTS TiposEmpleados;
```

```
CREATE TABLE TiposEmpleados (
```

```
    IdTipoEmpleado int Identity(1, 1) primary key not null,
```

```
    TipoEmpleado varchar(50) not null ,
```

```
    UltimaActualizacion datetime not null default GETDATE()
```

```
);
```

```
/*
```

```
    Departamento
```

```
*/
```

```
DROP TABLE if EXISTS Departamentos;
```

```
CREATE TABLE Departamentos (
```

```
    IdDepartamento int Identity(1, 1) primary key not null,
```

```
    Departamento varchar(50) not null,
```

```
    UltimaActualizacion datetime not null default GETDATE()
```

```
);
```

```
/*
```

```
    Clientes
```

```
*/
```

```
DROP TABLE if EXISTS Clientes;
```

```
CREATE TABLE Clientes (  
    IdPersona int not null,  
    IdTiposClientes int not null,  
    Usuario varchar(50) not null ,  
    Password varchar(255) not null,  
    Credito float default 0,  
    UltimaActualizacion datetime not null default GETDATE()  
);
```

```
/*
```

```
    TiposClientes
```

```
*/
```

```
DROP TABLE if EXISTS TiposClientes;
```

```
CREATE TABLE TiposClientes (  
    IdTiposClientes int Identity(1, 1) primary key not null,  
    TipoCliente varchar not null ,  
    UltimaActualizacion datetime not null default GETDATE()  
);
```

```
/*
```

```
    Ordenes
```

```
*/
```

```
DROP TABLE if EXISTS Ordenes;
```

```
CREATE TABLE Ordenes (  
    IdOrden int Identity(1, 1) primary key not null,  
    Codigo varchar(255) not null,  
    IdCliente int not null,  
    IdEmpleado int not null,  
    IdProducto int not null,  
    Cantidad int default 1,  
    UltimaActualizacion datetime not null default GETDATE()  
);
```


/*

Ventas

***/**

DROP TABLE if EXISTS Ventas;

CREATE TABLE Ventas (

CodigoOrdenes varchar(255) primary key,

IdCliente int not null,

IdEmpleado int not null,

SubTotal float not null,

ITBIS float not null,

Total float not null,

Efectivo float not null,

Cambio as Efectivo - Total,

UltimaActualizacion datetime not null default GETDATE()

);

/*

Inventario

***/**

DROP TABLE if EXISTS Inventario;

CREATE TABLE Inventario (

IdInventario int Identity(1, 1) primary key not null,

IdProducto int not null,

Stock int not null,

UltimaActualizacion datetime not null default GETDATE()

);

/*

Logs

***/**

DROP TABLE if EXISTS Logs;

CREATE TABLE Logs (

IdLogs int Identity(1, 1) primary key not null,

```
        IdUsuario int not null,  
        Log varchar(255) not null,  
        Fecha datetime not null default GETDATE()  
    );
```

```
/*
```

```
    4)      Query de creación de bases de datos, tablas, relaciones, vistas, procedimientos  
almacenados.
```

```
*/
```

```
--create database TecnoDigitalBD;
```

```
use TecnoDigitalBD;
```

```
--insert:
```

```
/*
```

```
    * TiposEmpleados
```

```
    * Departamentos
```

```
    * TiposProductos
```

```
*/
```

```
insert into TiposEmpleados
```

```
(TipoEmpleado)
```

```
Values
```

```
('TipoEmpleado1'),
```

```
('TipoEmpleado2'),
```

```
('TipoEmpleado3'),
```

```
('TipoEmpleado4'),
```

```
('TipoEmpleado5'),
```

```
('TipoEmpleado6'),
```

```
('TipoEmpleado7'),
```

```
('TipoEmpleado8'),
```

```
('TipoEmpleado9'),
```

```
('TipoEmpleado10');
```

```
insert into Departamentos
```

```
(Departamento)
```

```
Values
```

```
('Departamento1'),
```

```
('Departamento2'),  
('Departamento3'),  
('Departamento4'),  
('Departamento5'),  
('Departamento6'),  
('Departamento7'),  
('Departamento8'),  
('Departamento9'),  
('Departamento10');
```

```
insert into TiposProductos
```

```
(TipoProducto)
```

```
Values
```

```
('Leche y derivados'),  
('Carne pescados y huevos'),  
('Legumbres, frutos secos y patatas'),  
('Hortalizas. Alimentos reguladores'),  
('Frutas. Alimentos reguladores'),  
('Cereales. Alimentos energéticos'),  
('Mantecas y aceites'),  
('Leche y derivados2'),  
('Carne pescados y huevos2'),  
('Legumbres, frutos secos y patatas2');
```

```
--Procedimientos:
```

```
/*
```

```
    * Insertar Empleado
```

```
*/
```

```
drop procedure if exists ProclInsertarEmpleado
```

```
go
```

```
CREATE PROCEDURE ProclInsertarEmpleado
```

```
@Nombres varchar(50), @Apellidos varchar(50),
```

```

@Cedula varchar(50), @Pasaporte varchar(50),
@IdTipoEmpleado int, @IdDepartamento int, @Cargo varchar(50),
@Usuario varchar(50), @Password varchar(255)
AS
    BEGIN
        --Insertar Persona
        insert into Personas
        (Nombres, Apellidos, Cedula, Pasaporte)
        Values
        (@Nombres, @Apellidos, @Cedula, @Pasaporte);

        --Optener Ultimo Id Insertado de Personas
        DECLARE @UltimoldPersona int;
        set @UltimoldPersona = SCOPE_IDENTITY();

        --Insertar Empleado
        insert into Empleados
        (IdPersona, IdTipoEmpleado, IdDepartamento, Cargo, Usuario, Password)
        Values
        (@UltimoldPersona, @IdTipoEmpleado, @IdDepartamento, @Cargo, @Usuario,
@Password);

        --Optener Ultimo Id Insertado de Empleado
        DECLARE @UltimoldEmpleado int;
        set @UltimoldEmpleado = SCOPE_IDENTITY();

        Select @UltimoldEmpleado as 'UltimoldEmpleado', @UltimoldPersona as
'UltimoldPersona';
    end
go

exec ProclInsertarEmpleado 'Nombres1', 'Apellidos1', '402-4015107-1', null, 1, 1, 'Programador1',
'user1', '123';

exec ProclInsertarEmpleado 'Nombres2', 'Apellidos2', '402-4015107-2', null, 2, 2, 'Programador2',
'user2', '123';

exec ProclInsertarEmpleado 'Nombres3', 'Apellidos3', '402-4015107-3', null, 3, 3, 'Programador3',
'user3', '123';

```

```
exec ProInsertarEmpleado 'Nombres4', 'Apellidos4', '402-4015107-4', null, 4, 4, 'Programador4',  
'user4', '123';
```

```
exec ProInsertarEmpleado 'Nombres5', 'Apellidos5', '402-4015107-5', null, 5, 5, 'Programador5',  
'user5', '123';
```

```
exec ProInsertarEmpleado 'Nombres6', 'Apellidos6', '402-4015107-6', null, 6, 6, 'Programador6',  
'user6', '123';
```

```
exec ProInsertarEmpleado 'Nombres7', 'Apellidos7', '402-4015107-7', null, 7, 7, 'Programador7',  
'user7', '123';
```

```
exec ProInsertarEmpleado 'Nombres8', 'Apellidos8', '402-4015107-8', null, 8, 8, 'Programador8',  
'user8', '123';
```

```
exec ProInsertarEmpleado 'Nombres9', 'Apellidos9', '402-4015107-9', null, 9, 9, 'Programador9',  
'user9', '123';
```

```
exec ProInsertarEmpleado 'Nombres10', 'Apellidos10', '402-4015107-10', null, 10, 10,  
'Programador10', 'user10', '123';
```

```
/*
```

```
    * Insertar Cliente
```

```
*/
```

```
drop procedure if exists ProInsertarCliente
```

```
go
```

```
CREATE PROCEDURE ProInsertarCliente
```

```
    @Nombres varchar(50), @Apellidos varchar(50),
```

```
    @Cedula varchar(50), @Pasaporte varchar(50),
```

```
    @IdTiposClientes int, @Usuario varchar(50), @Password varchar(50)
```

```
AS
```

```
    BEGIN
```

```
        --Insertar Persona
```

```
        insert into Personas
```

```
            (Nombres, Apellidos, Cedula, Pasaporte)
```

```
        Values
```

```
            (@Nombres, @Apellidos, @Cedula, @Pasaporte);
```

```
        --Optener Ultimo Id Insertado de Personas
```

```
        DECLARE @UltimoldPersona int;
```

```
        set @UltimoldPersona = SCOPE_IDENTITY();
```

```

--Insertar Cliente
insert into Clientes
(IdPersona, IdTiposClientes, Usuario, Password)
Values
(@UltimoldPersona, @IdTiposClientes, @Usuario, @Password);

--Optener Ultimo Id Insertado de Clientes
DECLARE @UltimoldClientes int;
set @UltimoldClientes = SCOPE_IDENTITY();

Select @UltimoldClientes as 'UltimoldClientes', @UltimoldPersona as
'UltimoldPersona';

end

go

exec ProclInsertarCliente 'Nombres1', 'Apellidos1', '402-4015107-11', null, 1, 'user1', '123';
exec ProclInsertarCliente 'Nombres2', 'Apellidos2', '402-4015107-12', null, 1, 'user2', '123';
exec ProclInsertarCliente 'Nombres3', 'Apellidos3', '402-4015107-13', null, 1, 'user3', '123';
exec ProclInsertarCliente 'Nombres4', 'Apellidos4', '402-4015107-14', null, 1, 'user4', '123';
exec ProclInsertarCliente 'Nombres5', 'Apellidos5', '402-4015107-15', null, 1, 'user5', '123';
exec ProclInsertarCliente 'Nombres6', 'Apellidos6', '402-4015107-16', null, 1, 'user6', '123';
exec ProclInsertarCliente 'Nombres7', 'Apellidos7', '402-4015107-17', null, 1, 'user7', '123';
exec ProclInsertarCliente 'Nombres8', 'Apellidos8', '402-4015107-18', null, 1, 'user8', '123';
exec ProclInsertarCliente 'Nombres9', 'Apellidos9', '402-4015107-19', null, 1, 'user9', '123';
exec ProclInsertarCliente 'Nombres10', 'Apellidos10', '402-4015107-20', null, 1, 'user10', '123';

/*
* Insertar Productos
*/

drop procedure if exists ProclInsertarProductos

go

CREATE PROCEDURE ProclInsertarProductos
@IdTipoProducto int, @NombreProducto varchar(50),

```

@DescripcionProducto varchar(50), @Precio int

AS

BEGIN

--Insertar Productos

insert into Productos

(IdTipoProducto, NombreProducto, DescripcionProducto, Precio)

Values

(@IdTipoProducto, @NombreProducto, @DescripcionProducto, @Precio);

--Optener Ultimo Id Insertado de Productos

DECLARE @UltimoldProductos int;

set @UltimoldProductos = SCOPE_IDENTITY();

Select @UltimoldProductos as 'UltimoldProductos';

end

go

exec ProclInsertarProductos 1, 'Producto1', 'Esta es la descripción del producto1', 500;

exec ProclInsertarProductos 2, 'Producto2', 'Esta es la descripción del producto2', 500;

exec ProclInsertarProductos 3, 'Producto3', 'Esta es la descripción del producto3', 500;

exec ProclInsertarProductos 4, 'Producto4', 'Esta es la descripción del producto4', 500;

exec ProclInsertarProductos 5, 'Producto5', 'Esta es la descripción del producto5', 500;

exec ProclInsertarProductos 6, 'Producto6', 'Esta es la descripción del producto6', 500;

exec ProclInsertarProductos 7, 'Producto7', 'Esta es la descripción del producto7', 500;

exec ProclInsertarProductos 8, 'Producto8', 'Esta es la descripción del producto8', 500;

exec ProclInsertarProductos 9, 'Producto9', 'Esta es la descripción del producto9', 500;

exec ProclInsertarProductos 10, 'Producto10', 'Esta es la descripción del producto10', 500;

/*

*** Insertar Inventario**

***/**

drop procedure if exists ProcUpdateStockInventario

go

CREATE PROCEDURE ProcUpdateStockInventario

@IdProducto int, @Stock int

AS

BEGIN

--Insertar Productos

Update Inventario set Stock = @Stock where @IdProducto = @IdProducto;

Select 'Update Inventario';

end

go

exec ProcUpdateStockInventario 1, 'Producto1', 'Esta es la descripción del producto1', 500;

/*

*** Insertar Ordenes**

***/**

drop procedure if exists ProclInsertarOrdenes

go

CREATE PROCEDURE ProclInsertarOrdenes

@Codigo varchar(255), @IdCliente int,

@IdEmpleado int, @IdProducto int,

@Cantidad int

AS

BEGIN

--Insertar Ordenes

insert into Ordenes

(Codigo, IdCliente, IdEmpleado, IdProducto, Cantidad)

Values

(@Codigo, @IdCliente, @IdEmpleado, @IdProducto, @Cantidad);

--Optener Ultimo Id Insertado de Ordenes

DECLARE @UltimoldOrdenes int;

set @UltimoldOrdenes = SCOPE_IDENTITY();

Select @UltimoldOrdenes as 'UltimoldOrdenes';

end

go

declare @Codigo varchar(255);

set @Codigo = CONVERT(varchar(255), NEWID());

exec ProclInsertarOrdenes @Codigo, 1, 1, 1, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 2, 3;

exec ProclInsertarOrdenes @Codigo, 1, 1, 3, 2;

exec ProclInsertarOrdenes @Codigo, 1, 1, 4, 5;

exec ProclInsertarOrdenes @Codigo, 1, 1, 5, 6;

exec ProclInsertarOrdenes @Codigo, 1, 1, 6, 10;

exec ProclInsertarOrdenes @Codigo, 1, 1, 7, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 8, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 9, 2;

exec ProclInsertarOrdenes @Codigo, 1, 1, 10, 3;

/*

* Insertar Ventas

*/

drop procedure if exists ProclInsertarVentas

go

CREATE PROCEDURE ProclInsertarVentas

@CodigoOrdenes varchar(255),

@IdCliente int,

@IdEmpleado int,

@Efectivo int

AS

BEGIN

--Optener Ultimo Id Insertado de Ordenes

DECLARE @SubTotal float;

set @SubTotal = (

select sum(Productos.Precio * Ordenes.Cantidad)

from Ordenes

```

        inner join Productos
        on Ordenes.IdProducto = Productos.IdProducto
        where Ordenes.Codigo = @CodigoOrdenes
    );

    Declare @ITBIS int;
    set @ITBIS = @SubTotal * 0.18;

    Declare @Total int;
    set @Total = @SubTotal + @ITBIS;

    --Insertar Ordenes
    insert into Ventas
    (CodigoOrdenes, IdCliente, IdEmpleado, SubTotal, ITBIS, Total, Efectivo)
    Values
    (@CodigoOrdenes, @IdCliente, @IdEmpleado, @SubTotal, @ITBIS, @Total,
    @Efectivo);

    --Optener Ultimo Id Insertado de Ordenes
    DECLARE @UltimoldOrdenes int;
    set @UltimoldOrdenes = SCOPE_IDENTITY();

    Select top(1) * from Ventas where CodigoOrdenes = @CodigoOrdenes order by
    CodigoOrdenes desc;
    end
go

exec ProcInsertarVentas '@CodigoOrdenes', 1, 1, 100000;

/*
    d) Actualizar cantidad en existencia de un producto con registro en tabla
    modificaciones de quien fue que lo modifiko.
*/
drop procedure if exists ProcNuevoLog
go

CREATE PROCEDURE ProcNuevoLog

```

@IdUsuario int, @Log Varchar(255)

AS

BEGIN

--Insert Log

insert into Logs

(IdUsuario, Log)

Values

(@IdUsuario, @Log);

--Optener Ultimo Id Insertado de Ordenes

DECLARE @UltimoldLog int;

set @UltimoldLog = SCOPE_IDENTITY();

Select @UltimoldLog as 'UltimoldLog';

end

go

exec ProcNuevoLog 1, 'test';

select * from Inventario;

/*

7) Realizar las siguientes vistas:

- a) Listar ordenes con su detalle**
- b) Informacion de clientes**
- c) Informacion de empleados(si aplica)**
- d) Clientes con más pedidos**

***/**

/*

a) Listar ordenes con su detalle

*/

Drop view if EXISTS VistaListaOrdenes;

go

create view VistaListaOrdenes as (

Select

Ordenes.Codigo,

Ordenes.IdCliente,

Ordenes.IdEmpleado,

Ordenes.IdProducto,

Ordenes.Cantidad,

Ventas.SubTotal,

Ventas.ITBIS,

Ventas.Total,

Ventas.Efectivo,

Ventas.Cambio

from Ventas

inner join Ordenes

on Ventas.CodigoOrdenes = Ordenes.Codigo

);

go

select * from VistaListaOrdenes;

/*

b) Informacion de clientes

*/

Drop view if EXISTS VistalInfoClientes;

go

create view VistalInfoClientes as (

Select

Clientes.IdTiposClientes,

Clientes.Usuario,

```
        Personas.Nombres,  
        Personas.Apellidos,  
        Personas.Cedula,  
        Personas.Pasaporte  
    from Clientes  
    inner join Personas  
    on Clientes.IdPersona = Personas.IdPersona  
);  
go
```

```
select * from VistaInfoClientes;
```

```
/*  
    c)      Informacion de empleados(si aplica)  
*/  
Drop view if EXISTS VistaInfoEmpleados;  
go
```

```
create view VistaInfoEmpleados as (  
    Select  
        Empleados.IdTipoEmpleado,  
        Empleados.Usuario,  
        Personas.Nombres,  
        Personas.Apellidos,  
        Personas.Cedula,  
        Personas.Pasaporte  
    from Empleados  
    inner join Personas  
    on Empleados.IdPersona = Personas.IdPersona  
);  
go
```

```
select * from VistaInfoEmpleados;
```

```
/*  
    d)      Clientes con más pedidos
```

***/**

Drop view if EXISTS VistaClientesOrdenes

go

create view VistaClientesOrdenes as (

Select COUNT(Ordenes.Codigo) as 'cantidad',

Ventas.CodigoOrdenes

from Ordenes

inner join Ventas

on Ordenes.Codigo = Ventas.CodigoOrdenes

group by Ventas.CodigoOrdenes

);

go

select * from VistaClientesOrdenes;

5) Crear tablas necesarias para las operaciones básicas del negocio (registros de empleados, productos, ventas, inventario, clientes, entre otras). Mantener las bases de datos normalizadas hasta la 2da forma normal (presentar capturas del proceso de normalización)

6) Insertar mínimo 10 registros en las tablas que apliquen

7) Realizar las siguientes vistas:

- a) Listar ordenes con su detalle**
- b) Informacion de clientes**
- c) Informacion de empleados(si aplica)**
- d) Clientes con más pedidos**

/*

7) Realizar las siguientes vistas:

- a) Listar ordenes con su detalle**
- b) Informacion de clientes**
- c) Informacion de empleados(si aplica)**
- d) Clientes con más pedidos**

***/**

/*

a) Listar ordenes con su detalle

***/**

Drop view if EXISTS VistaListaOrdenes;

go

create view VistaListaOrdenes as (

Select

Ordenes.Codigo,

Ordenes.IdCliente,

Ordenes.IdEmpleado,

Ordenes.IdProducto,

Ordenes.Cantidad,

Ventas.SubTotal,

Ventas.ITBIS,

Ventas.Total,

Ventas.Efectivo,

Ventas.Cambio

from Ventas

inner join Ordenes

on Ventas.CodigoOrdenes = Ordenes.Codigo

);

go

select * from VistaListaOrdenes;

/*

b) Informacion de clientes

***/**

Drop view if EXISTS VistaInfoClientes;

go

```

create view VistaInfoClientes as (
    Select
        Clientes.IdTiposClientes,
        Clientes.Usuario,
        Personas.Nombres,
        Personas.Apellidos,
        Personas.Cedula,
        Personas.Pasaporte
    from Clientes
    inner join Personas
    on Clientes.IdPersona = Personas.IdPersona
);
go

```

```

select * from VistaInfoClientes;

```

```

/*
    c)      Informacion de empleados(si aplica)
*/
Drop view if EXISTS VistaInfoEmpleados;
go

```

```

create view VistaInfoEmpleados as (
    Select
        Empleados.IdTipoEmpleado,
        Empleados.Usuario,
        Personas.Nombres,
        Personas.Apellidos,
        Personas.Cedula,
        Personas.Pasaporte
    from Empleados
    inner join Personas
    on Empleados.IdPersona = Personas.IdPersona
);
go

```



```
select * from VistaInfoEmpleados;
```

```
/*
```

```
    d)    Clientes con más pedidos
```

```
*/
```

```
Drop view if EXISTS VistaClientesOrdenes
```

```
go
```

```
create view VistaClientesOrdenes as (
```

```
    Select COUNT(Ordenes.Codigo) as 'cantidad',
```

```
    Ventas.CodigoOrdenes
```

```
from Ordenes
```

```
inner join Ventas
```

```
on Ordenes.Codigo = Ventas.CodigoOrdenes
```

```
group by Ventas.CodigoOrdenes
```

```
);
```

```
go
```

```
select * from VistaClientesOrdenes;
```

8) Crear los siguientes procedimientos almacenados

a) Insertar registro de venta

b) Insertar empleado

c) Insertar producto

d) Actualizar cantidad en existencia de un producto con registro en tabla modificaciones de quien fue que lo modifico.

e) Filtrar cliente por ciudad.

```
/*
```

```
    * Insertar Empleado
```

```
*/
```

```
drop procedure if exists ProclInsertarEmpleado
```

```
go
```

```
CREATE PROCEDURE ProclInsertarEmpleado
```

```

@Nombres varchar(50), @Apellidos varchar(50),
@Cedula varchar(50), @Pasaporte varchar(50),
@IdTipoEmpleado int, @IdDepartamento int, @Cargo varchar(50),
@Usuario varchar(50), @Password varchar(255)
AS
    BEGIN
        --Insertar Persona
        insert into Personas
        (Nombres, Apellidos, Cedula, Pasaporte)
        Values
        (@Nombres, @Apellidos, @Cedula, @Pasaporte);

        --Optener Ultimo Id Insertado de Personas
        DECLARE @UltimoldPersona int;
        set @UltimoldPersona = SCOPE_IDENTITY();

        --Insertar Empleado
        insert into Empleados
        (IdPersona, IdTipoEmpleado, IdDepartamento, Cargo, Usuario, Password)
        Values
        (@UltimoldPersona, @IdTipoEmpleado, @IdDepartamento, @Cargo, @Usuario,
@Password);

        --Optener Ultimo Id Insertado de Empleado
        DECLARE @UltimoldEmpleado int;
        set @UltimoldEmpleado = SCOPE_IDENTITY();

        Select  @UltimoldEmpleado as 'UltimoldEmpleado', @UltimoldPersona as
'UltimoldPersona';
    end
go

exec ProclInsertarEmpleado 'Nombres1', 'Apellidos1', '402-4015107-1', null, 1, 1, 'Programador1',
'user1', '123';

exec ProclInsertarEmpleado 'Nombres2', 'Apellidos2', '402-4015107-2', null, 2, 2, 'Programador2',
'user2', '123';

exec ProclInsertarEmpleado 'Nombres3', 'Apellidos3', '402-4015107-3', null, 3, 3, 'Programador3',
'user3', '123';

```

```
exec ProInsertarEmpleado 'Nombres4', 'Apellidos4', '402-4015107-4', null, 4, 4, 'Programador4',  
'user4', '123';
```

```
exec ProInsertarEmpleado 'Nombres5', 'Apellidos5', '402-4015107-5', null, 5, 5, 'Programador5',  
'user5', '123';
```

```
exec ProInsertarEmpleado 'Nombres6', 'Apellidos6', '402-4015107-6', null, 6, 6, 'Programador6',  
'user6', '123';
```

```
exec ProInsertarEmpleado 'Nombres7', 'Apellidos7', '402-4015107-7', null, 7, 7, 'Programador7',  
'user7', '123';
```

```
exec ProInsertarEmpleado 'Nombres8', 'Apellidos8', '402-4015107-8', null, 8, 8, 'Programador8',  
'user8', '123';
```

```
exec ProInsertarEmpleado 'Nombres9', 'Apellidos9', '402-4015107-9', null, 9, 9, 'Programador9',  
'user9', '123';
```

```
exec ProInsertarEmpleado 'Nombres10', 'Apellidos10', '402-4015107-10', null, 10, 10,  
'Programador10', 'user10', '123';
```

```
/*
```

```
    * Insertar Cliente
```

```
*/
```

```
drop procedure if exists ProInsertarCliente
```

```
go
```

```
CREATE PROCEDURE ProInsertarCliente
```

```
    @Nombres varchar(50), @Apellidos varchar(50),
```

```
    @Cedula varchar(50), @Pasaporte varchar(50),
```

```
    @IdTiposClientes int, @Usuario varchar(50), @Password varchar(50)
```

```
AS
```

```
    BEGIN
```

```
        --Insertar Persona
```

```
        insert into Personas
```

```
            (Nombres, Apellidos, Cedula, Pasaporte)
```

```
        Values
```

```
            (@Nombres, @Apellidos, @Cedula, @Pasaporte);
```

```
        --Optener Ultimo Id Insertado de Personas
```

```
        DECLARE @UltimoldPersona int;
```

```
        set @UltimoldPersona = SCOPE_IDENTITY();
```

```

--Insertar Cliente
insert into Clientes
(IdPersona, IdTiposClientes, Usuario, Password)
Values
(@UltimoldPersona, @IdTiposClientes, @Usuario, @Password);

--Optener Ultimo Id Insertado de Clientes
DECLARE @UltimoldClientes int;
set @UltimoldClientes = SCOPE_IDENTITY();

Select  @UltimoldClientes as 'UltimoldClientes', @UltimoldPersona as
'UltimoldPersona';

end

go

exec ProInsertarCliente 'Nombres1', 'Apellidos1', '402-4015107-11', null, 1, 'user1', '123';
exec ProInsertarCliente 'Nombres2', 'Apellidos2', '402-4015107-12', null, 1, 'user2', '123';
exec ProInsertarCliente 'Nombres3', 'Apellidos3', '402-4015107-13', null, 1, 'user3', '123';
exec ProInsertarCliente 'Nombres4', 'Apellidos4', '402-4015107-14', null, 1, 'user4', '123';
exec ProInsertarCliente 'Nombres5', 'Apellidos5', '402-4015107-15', null, 1, 'user5', '123';
exec ProInsertarCliente 'Nombres6', 'Apellidos6', '402-4015107-16', null, 1, 'user6', '123';
exec ProInsertarCliente 'Nombres7', 'Apellidos7', '402-4015107-17', null, 1, 'user7', '123';
exec ProInsertarCliente 'Nombres8', 'Apellidos8', '402-4015107-18', null, 1, 'user8', '123';
exec ProInsertarCliente 'Nombres9', 'Apellidos9', '402-4015107-19', null, 1, 'user9', '123';
exec ProInsertarCliente 'Nombres10', 'Apellidos10', '402-4015107-20', null, 1, 'user10', '123';

/*
* Insertar Productos
*/

drop procedure if exists ProInsertarProductos

go

CREATE PROCEDURE ProInsertarProductos
@IdTipoProducto int, @NombreProducto varchar(50),

```

@DescripcionProducto varchar(50), @Precio int

AS

BEGIN

--Insertar Productos

insert into Productos

(IdTipoProducto, NombreProducto, DescripcionProducto, Precio)

Values

(@IdTipoProducto, @NombreProducto, @DescripcionProducto, @Precio);

--Optener Ultimo Id Insertado de Productos

DECLARE @UltimoldProductos int;

set @UltimoldProductos = SCOPE_IDENTITY();

Select @UltimoldProductos as 'UltimoldProductos';

end

go

exec ProclInsertarProductos 1, 'Producto1', 'Esta es la descripción del producto1', 500;

exec ProclInsertarProductos 2, 'Producto2', 'Esta es la descripción del producto2', 500;

exec ProclInsertarProductos 3, 'Producto3', 'Esta es la descripción del producto3', 500;

exec ProclInsertarProductos 4, 'Producto4', 'Esta es la descripción del producto4', 500;

exec ProclInsertarProductos 5, 'Producto5', 'Esta es la descripción del producto5', 500;

exec ProclInsertarProductos 6, 'Producto6', 'Esta es la descripción del producto6', 500;

exec ProclInsertarProductos 7, 'Producto7', 'Esta es la descripción del producto7', 500;

exec ProclInsertarProductos 8, 'Producto8', 'Esta es la descripción del producto8', 500;

exec ProclInsertarProductos 9, 'Producto9', 'Esta es la descripción del producto9', 500;

exec ProclInsertarProductos 10, 'Producto10', 'Esta es la descripción del producto10', 500;

/*

*** Insertar Inventario**

***/**

drop procedure if exists ProcUpdateStockInventario

go

CREATE PROCEDURE ProcUpdateStockInventario

@IdProducto int, @Stock int

AS

BEGIN

--Insertar Productos

Update Inventario set Stock = @Stock where @IdProducto = @IdProducto;

Select 'Update Inventario';

end

go

exec ProcUpdateStockInventario 1, 'Producto1', 'Esta es la descripción del producto1', 500;

/*

*** Insertar Ordenes**

***/**

drop procedure if exists ProclInsertarOrdenes

go

CREATE PROCEDURE ProclInsertarOrdenes

@Codigo varchar(255), @IdCliente int,

@IdEmpleado int, @IdProducto int,

@Cantidad int

AS

BEGIN

--Insertar Ordenes

insert into Ordenes

(Codigo, IdCliente, IdEmpleado, IdProducto, Cantidad)

Values

(@Codigo, @IdCliente, @IdEmpleado, @IdProducto, @Cantidad);

--Optener Ultimo Id Insertado de Ordenes

DECLARE @UltimoldOrdenes int;

set @UltimoldOrdenes = SCOPE_IDENTITY();

Select @UltimoldOrdenes as 'UltimoldOrdenes';

end

go

declare @Codigo varchar(255);

set @Codigo = CONVERT(varchar(255), NEWID());

exec ProclInsertarOrdenes @Codigo, 1, 1, 1, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 2, 3;

exec ProclInsertarOrdenes @Codigo, 1, 1, 3, 2;

exec ProclInsertarOrdenes @Codigo, 1, 1, 4, 5;

exec ProclInsertarOrdenes @Codigo, 1, 1, 5, 6;

exec ProclInsertarOrdenes @Codigo, 1, 1, 6, 10;

exec ProclInsertarOrdenes @Codigo, 1, 1, 7, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 8, 1;

exec ProclInsertarOrdenes @Codigo, 1, 1, 9, 2;

exec ProclInsertarOrdenes @Codigo, 1, 1, 10, 3;

/*

* Insertar Ventas

*/

drop procedure if exists ProclInsertarVentas

go

CREATE PROCEDURE ProclInsertarVentas

@CodigoOrdenes varchar(255),

@IdCliente int,

@IdEmpleado int,

@Efectivo int

AS

BEGIN

--Optener Ultimo Id Insertado de Ordenes

DECLARE @SubTotal float;

set @SubTotal = (

select sum(Productos.Precio * Ordenes.Cantidad)

from Ordenes

```

        inner join Productos
        on Ordenes.IdProducto = Productos.IdProducto
        where Ordenes.Codigo = @CodigoOrdenes
    );

    Declare @ITBIS int;
    set @ITBIS = @SubTotal * 0.18;

    Declare @Total int;
    set @Total = @SubTotal + @ITBIS;

    --Insertar Ordenes
    insert into Ventas
    (CodigoOrdenes, IdCliente, IdEmpleado, SubTotal, ITBIS, Total, Efectivo)
    Values
    (@CodigoOrdenes, @IdCliente, @IdEmpleado, @SubTotal, @ITBIS, @Total,
    @Efectivo);

    --Optener Ultimo Id Insertado de Ordenes
    DECLARE @UltimoldOrdenes int;
    set @UltimoldOrdenes = SCOPE_IDENTITY();

    Select top(1) * from Ventas where CodigoOrdenes = @CodigoOrdenes order by
    CodigoOrdenes desc;
end
go

exec ProcInsertarVentas '@CodigoOrdenes', 1, 1, 100000;

/*
    d) Actualizar cantidad en existencia de un producto con registro en tabla
    modificaciones de quien fue que lo modifiko.
*/
drop procedure if exists ProcNuevoLog
go

CREATE PROCEDURE ProcNuevoLog

```


@IdUsuario int, @Log Varchar(255)

AS

BEGIN

--Insert Log

insert into Logs

(IdUsuario, Log)

Values

(@IdUsuario, @Log);

--Optener Ultimo Id Insertado de Ordenes

DECLARE @UltimoldLog int;

set @UltimoldLog = SCOPE_IDENTITY();

Select @UltimoldLog as 'UltimoldLog';

end

go

exec ProcNuevoLog 1, 'test';