



BASE  
DE  
DATOS I

NOVIEMBRE 2021

# Proyecto Final Base De Datos I

ESTUDIANTE

José Félix  
Sarmiento Mamani  
5159119638

[CLICK AQUÍ](#)  
[PARA ACCEDER AL](#)  
[REPOSITORIO](#)

DOCENTE

William Roddy  
Barra Paredes



# 1. Introducción.

Especificar cuál es el objetivo del proyecto.

*El objetivo del proyecto es tener un control de inventario de la empresa XYZ. Asi mismo que esta pueda realizar una compra generando una factura y el registro del cliente que realizó la compra.*

The screenshot shows a software window titled "Venta - Empresa XYZ" with a menu bar containing "Ajustes", "Usuarios", "Inventario", and "Registro inventario". The main form includes a "No Factura:" field with the value "2" and a "Cliente:" field. Below these is a table with columns: "Codigo", "Producto", "Precio x Unidad", "Cantidad", and "Descuento". The table has one row with an asterisk (\*) in the first column. To the left of the table is a "NIT Cliente:" field and a "Buscar" button. At the bottom, there are fields for "SUBTOTAL" (0.00), "Imp. Venta" (9 %), "Descuento" (0 %), and "TOTAL" (0.00). There are also buttons for "Vista Previa", "Realizar compra", "Agregar producto", and "Cerrar".

	Codigo	Producto	Precio x Unidad	Cantidad	Descuento
*					

Subtotal: 0.00  
Imp. Venta: 9 %  
Descuento: 0 %  
Total: 0.00

## 2. Diseño de la base de Datos.

### 2.1. Contexto de la Base de Datos.

*Dada la siguiente situación del software se requiere: registro del personal que pueda acceder al software, registro del todo el personal de la empresa, registro de los clientes, registro del inventario, registro de datos de facturación, todo esto en una BBDD de la empresa XYZ*

### 2.2. Análisis y definición de Tablas.

Se requiere una tabla que tenga datos de los usuarios que pueda **logearse** para acceder al sistema de ventas. Se requiere otra tabla que contenga **datos del personal** de la empresa XYZ así también como su CI, que pueda referenciarse a la tabla de usuarios con login. Otra tabla que tenga datos del **inventario**, que contenga el código del producto, stock, y nombre del mismo. También se requiere una tabla donde se almacenen los datos del **cliente**. Se requiere una tabla de **facturación** que contenga el nombre del producto y numero factura.

Ejemplo:

NOMBRE DE LA TABLA	Descripción
login	Almacena datos de los usuarios que pueda logearse
personas	Almacena datos personales de todo el personal de la empresa
inventario	Esta tabla almacena de todos los productos que se encuentra disponibles
cliente	Almacena los datos de los clientes
facturación	Guarda el n de factura y los productos comprados por el cliente

2.3. Diseño de la Base de Datos.

2.3.1 Código SQL de las tablas.

Generar el código SQL de las tablas.

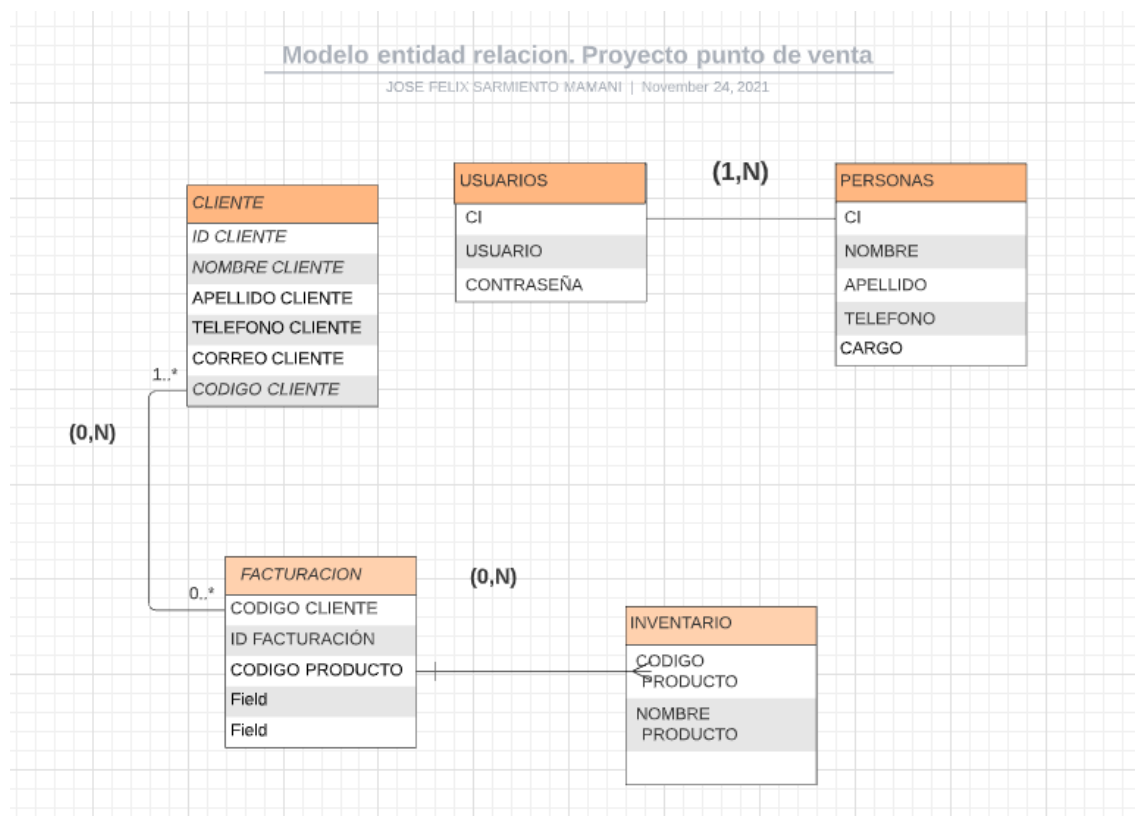
Ejemplo:

LOGIN	<code>create table</code> usuarioPrueba (
-------	--

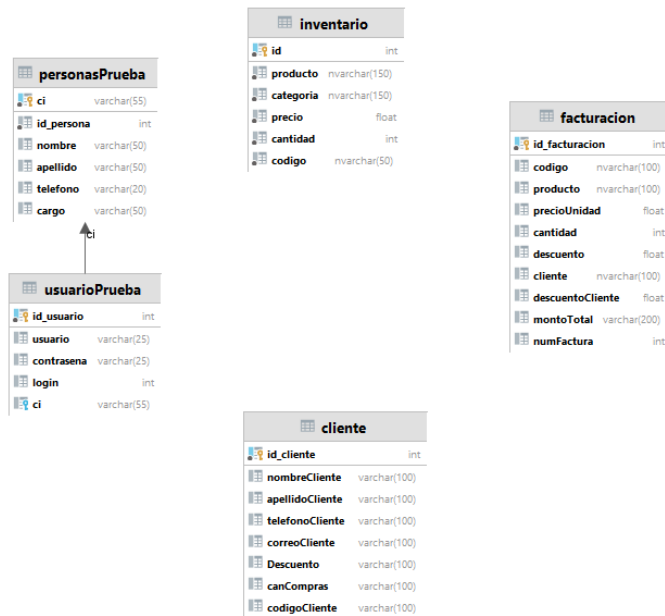
	<pre> id_usuario int identity     primary key, usuario    varchar(25), contrasena varchar(25), login      int, ci          varchar(55)     references personasPrueba ) </pre>
PERSONAS	<pre> create table personas (     id_persona int identity         primary key, nombre        varchar(50), apellido      varchar(50), ci            varchar(25) not null, telefono      varchar(20), usuario       varchar(25), contrasena    varchar(25), login         int ) </pre>
INVENTARIO	<pre> create table inventario (     id            int identity         primary key, producto         nvarchar(150) not null, categoria        nvarchar(150) not null, precio           float          not null, cantidad         int            not null, codigo           nvarchar(50)   not null ) </pre>
CLIENTE	<pre> create table cliente (     id_cliente      int identity         primary key, nombreCliente     varchar(100), apellidoCliente   varchar(100), telefonoCliente   varchar(100), correoCliente     varchar(100), Descuento         varchar(100), canCompras        varchar(100), codigoCliente     varchar(100) ) </pre>
FACTURACIÓN	<pre> create table facturacion (     id_facturacion int identity         primary key, </pre>

	<pre> codigo          nvarchar(100), producto        nvarchar(100), precioUnidad    float, cantidad        int, descuento       float, cliente         nvarchar(100), descuentoCliente float, montoTotal      varchar(200), numFactura      int         ) </pre>
--	--

### 2.3.2 Modelo entidad relación de la Base de Datos ER.



### 2.3.3 Modelo lógico de la Base de Datos.

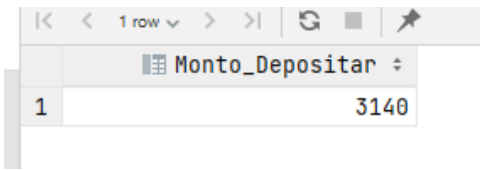


### 2.4. Búsquedas, funciones, vistas, etc.

La base de datos mas allá de tener las tablas y los registros correspondientes debe de tener vistas, funciones, etc.

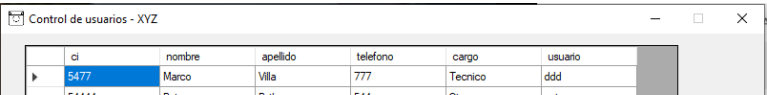
<pre> CREATE OR ALTER VIEW VistaGeneral AS SELECT per.nombre, per.apellido, per.telefono, per.cargo, usu.usuario, usu.ci, usu.login FROM personas AS per left join usuarios AS usu on per.ci = usu.ci; </pre>	
<pre> CREATE OR ALTER VIEW General_Factura AS SELECT numFactura, usu.usuario AS Vendedor, fac.cliente, fac.fecha, sum(convert(int, fac.cantidad)) AS Cantidad_Productos, sum(convert(int, fac.montoTotal) ) AS Total_a_Pagar FROM facturacion AS fac INNER JOIN usuarios AS usu ON fac.id_usuario = usu.id_usuario GROUP BY numFactura, fac.cliente, fac.fecha, usu.usuario </pre>	
<pre> CREATE OR ALTER FUNCTION dbo.ventasUsuario (@val VARCHAR(50)) RETURNS TABLE AS return (SELECT cli.nombreCliente AS Cliente, fact.numFactura, fact.producto AS Producto, fact.precioUnidad AS PrecioUnidad, fact.cantidad as Cantidad, fact.montoTotal as PrecioTotal, usu.usuario as Usuario </pre>	

	<pre>FROM facturacion AS fact INNER JOIN usuarios AS usu on usu.id_usuario = fact.id_usuario INNER JOIN cliente as cli on cli.id_cliente = fact.id_cliente where usu.ci=@val);</pre>																		
<table><tr><th>Cliente</th><th>Producto</th><th>PrecioUnidad</th><th>Cantidad</th><th>PrecioTotal</th><th>Usuario</th></tr><tr><td>Ing</td><td>RJ45 CABLE</td><td>100</td><td>3</td><td>300</td><td>Carlos</td></tr><tr><td>Ing</td><td>HP L45 TON</td><td>100</td><td>1</td><td>100</td><td>Carlos</td></tr></table>	Cliente	Producto	PrecioUnidad	Cantidad	PrecioTotal	Usuario	Ing	RJ45 CABLE	100	3	300	Carlos	Ing	HP L45 TON	100	1	100	Carlos	<pre>CREATE OR ALTER FUNCTION dbo.rastrearFactura (@valor INT) returns table as return (SELECT cli.nombreCliente AS Cliente, fact.producto AS Producto,fact.precioUnidad AS PrecioUnidad,fact.cantidad as Cantidad, fact.montoTotal as PrecioTotal,usu.usuario as Usuario FROM facturacion as fact INNER JOIN usuarios AS usu on usu.id_usuario = fact.id_usuario INNER JOIN cliente as cli on cli.id_cliente = fact.id_cliente WHERE numFactura=@valor); CREATE OR ALTER FUNCTION dbo.RasComprasCliente (@valor INT) returns table as return (SELECT cli.nombreCliente AS Cliente, fact.numFactura,fact.producto AS Producto,fact.precioUnidad AS PrecioUnidad,fact.cantidad as Cantidad, fact.montoTotal as PrecioTotal,usu.usuario as Usuario FROM facturacion as fact INNER JOIN usuarios AS usu on usu.id_usuario = fact.id_usuario INNER JOIN cliente as cli on cli.id_cliente = fact.id_cliente WHERE cli.codigoCliente=@valor);</pre>
Cliente	Producto	PrecioUnidad	Cantidad	PrecioTotal	Usuario														
Ing	RJ45 CABLE	100	3	300	Carlos														
Ing	HP L45 TON	100	1	100	Carlos														
<table><tr><th colspan="2">&lt;anonymous&gt;</th></tr><tr><td>1</td><td>2</td></tr></table>	<anonymous>		1	2	<pre>CREATE OR ALTER FUNCTION CantidadVentasDelDia () RETURNS INT AS BEGIN DECLARE @var INT SET @var=(SELECT COUNT(*) FROM facturacion INNER JOIN usuarios AS usu on usu.id_usuario = facturacion.id_usuario WHERE fecha= (SELECT CONVERT(VARCHAR(10), GETDATE(), 103)) ) RETURN @var; END ;</pre>														
<anonymous>																			
1	2																		
<table><tr><th colspan="2">&lt;anonymous&gt;</th></tr><tr><td>1</td><td>13</td></tr></table>	<anonymous>		1	13	<pre>CREATE OR ALTER FUNCTION CantidadArticulosVendedorPorUUsuario (@ci VARCHAR(20)) RETURNS INT AS BEGIN DECLARE @var INT SET @var=(SELECT SUM(cantidad) FROM facturacion INNER JOIN usuarios AS usu on usu.id_usuario = facturacion.id_usuario WHERE usu.ci =@ci) RETURN @var; END ;</pre>														
<anonymous>																			
1	13																		

	<pre> CREATE OR ALTER FUNCTION DineroUsuarioFecha (@ciUsuario int,@fecha varchar(10)) RETURNS INT AS BEGIN DECLARE @var INT SET @var=(SELECT SUM(convert(int,montoTotal)) AS Monto_Total FROM facturacion INNER JOIN usuarios u on u.id_usuario = facturacion.id_usuario INNER JOIN personas p on p.ci = u.ci WHERE p.ci=@ciUsuario AND fecha =@fecha) RETURN @var ; END; SELECT dbo.DineroUsuarioFecha ('ciUsuario ','FechaVentas') SELECT dbo.DineroUsuarioFecha ('8745111','11/11/2021') AS Monto_Depositar </pre>
---	---

### 3. Usabilidad

#### 3.1. Imágenes acerca del uso del sistema.



ci	nombre	apellido	telefono	cargo	usuario
5477	Marco	Villa	777	Tecnico	ddd
54444	Pato	Python	544	Otro	pato
444444	Alberto	Widdlston	544441	Administrador	Albert
54123	Antonio	Villarán	+591 7561874	Administrador	Mark
9119638	José Félix	Samiento Mamani	60580189	logistica	Felicks
5421789	Antonio	Garcia	54182	POR DEFECTO??	
*					

#### 3.2. Video acerca del uso del sistema.

Adjuntar video acerca del uso del sistema

[\[https://drive.google.com/file/d/10gEQ6po-Urvgf-cl\\_1PWw4LXYJDA8G6F/view?usp=sharing\]](https://drive.google.com/file/d/10gEQ6po-Urvgf-cl_1PWw4LXYJDA8G6F/view?usp=sharing)



## 4. Conclusión

Este proyecto me ayudó bastante al uso de la lógica de estructuración de un sistema en cuanto a BBDD. Se logró obtener una visualización más real de la aplicación de un sistema así también las necesidades que se puede tener al momento de requerir ciertos datos.

### Notas a considerarse:

- En **github** crear una nueva carpeta de nombre **PROYECTO\_FINAL**.
- Los documentos que deben estar presente en github son los siguientes.
  - Informe (documento solicitado en esta documentación)
  - Un archivo **excel**(u otro) con todos los participantes (el título del archivo debe ser el nombre del grupo)
  - Diagrama entidad relación (Imagen).
  - Diagrama modelo lógico (Imagen).
  - Archivo .sql con todo el código sql de la Base de Datos.