

Proyecto bases de datos

Diseño Papelería

Jiménez García María Fernanda

García Cruz Diana Aide

Brandon Silva Barrera

Paul Jaime Félix Flores

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

BASES DE DATOS

Fecha de entrega: 27 de mayo de 2020

Contents

1	Introducción	3
1.1	Objetivo	3
1.2	Descripción del análisis del problema	4
2	Plan de trabajo	4
3	Diseño	5
3.1	Análisis general	5
3.2	Modelo Entidad-Relación.	6
3.3	Modelo Relacional	9
3.4	Normalización	10
4	Implementación	13
4.1	Base de datos	13
4.2	Creación de tablas	14
4.3	Agregado de información	18
4.4	Consultas	22
5	Presentación	27
6	Conclusiones	28

Proyecto base de datos: Implementación de papelería

1 Introducción

El presente documento muestra el desarrollo del análisis, diseño y propuesta de implementación de una base de datos de para el almacenamiento de información de una papelería, de esta manera ayudará a la empresa a organizar, controlar y administrar los productos y las ventas, mejorando la interacción con los clientes.

1.1 Objetivo

Cada vez más empresas optan por implementar sistemas de gestión de ventas utilizando las tecnologías de información con el objetivo de incrementar los clientes y potenciales clientes, además de las ventas.

Por lo tanto, para el caso de una empresa de ventas de productos de papelería, se desarrolla un sistema de tienda virtual para la gestión de ventas que permita a los clientes registrarse, seleccionar los productos, comprarlos y pagarlos.

Implementar un sistema de tienda virtual genera diversos beneficios como: Manejar un registro de clientes, manejar un registro de ventas, permitir emitir reportes de ventas, clientes y productos actualizados que ayudan a la empresa a tomar mejores decisiones a corto y a largo plazo.

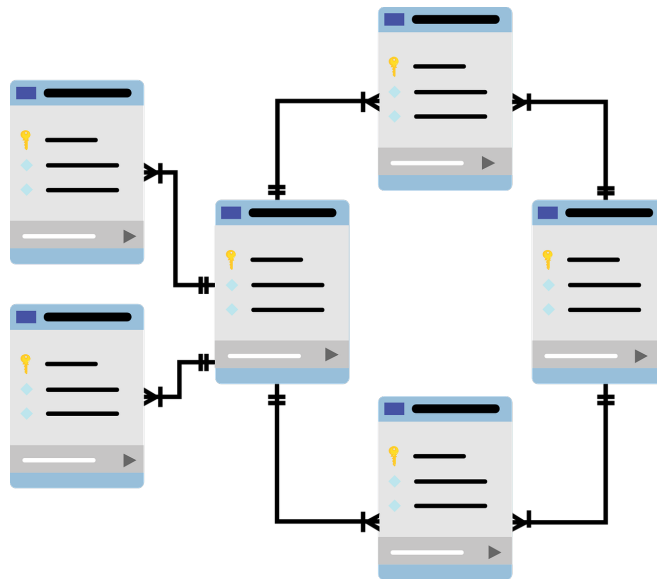


Figure 1: Diseño de base de datos

1.2 Descripción del análisis del problema

El proyecto requirió realizar un sistema informático para la administración de una papelería, de manera que los datos cambien en tiempo real y se mantengan consistentes siguiendo ciertas reglas de negocio. Para hacer lo anterior posible se siguieron las siguientes condiciones.

- El sistema maneja el inventariado de los productos almacenando sus características principales y su proveedor, a la vez que mantiene sus existencias(stock).
- Tiene un registro de los proveedores y su información.
- Almacena información de los clientes.
- Hace posible un proceso de ventas, de manera que en la venta registre el cliente, los productos que la conforman y un identificador único. Los productos vendidos se deben retirar del inventario. Además se debe calcular el monto total y el monto por cada artículo.

2 Plan de trabajo

Primero se hizo un análisis de los componentes con los que ya contaban dos equipos que conformaron el actual.

Se decidió adaptar los componentes de ambos con la finalidad de unir sus respectivos trabajos. Así, dos personas se encargaron del diseño de la base de datos y dos de la interfaz gráfica.

- *García Cruz Diana Aide*

Diseño e implementación de la base de datos:

Diseño de modelos entidad-relación y modelo relacional. Normalización de base de datos. Implementación en lenguaje SQL de base de datos y tablas. Participación en agregado de información. Implementación de consultas y funciones.

- *Jiménez García María Fernanda*

Diseño e implementación de la base de datos:

Diseño de modelos entidad-relación y modelo relacional. Normalización de base de datos. Implementación en lenguaje SQL de base de datos y tablas. Participación en agregado de información. Implementación de consultas y funciones.

Documentación: Redactada en latex.

- *Brandon Silva Barrera*

Participó en la documentación y unión de los elementos de ambos equipos que se decidió reutilizar, además que actuó como elemento de apoyo en otras áreas.

- *Paul Jaime Félix Flores*

Implementación de interfaz gráfica en java. Adaptación de la interfaz de usuario en java a la base de datos.

3 Diseño

Enunciado: Una cadena de papelerías busca innovar la manera en que almacena su información, y los

contratan para que desarrollen los sistemas informáticos para satisfacer los siguientes requerimientos:

Se desea tener almacenados datos como la razón social, domicilio, nombre y teléfonos de los proveedores, razón social, nombre, domicilio y al menos un email de los clientes. Es necesario tener un inventario de los productos que se venden, en el que debe guardarse el código de barras, precio al que fue comprado el producto, fecha de compra y cantidad de ejemplares en la bodega (stock).

Se desea guardar la marca, descripción y precio de los regalos, artículos de papelería, impresiones y recargas, siempre y cuando se tenga su correspondiente registro en el inventario. Debe también guardarse el número de venta, fecha de venta y la cantidad total a pagar de la venta, así como la cantidad de cada artículo y precio total a pagar por artículo. Además, se requiere que:

- Al recibir el código de barras de un producto, regrese la utilidad.
- Cada que haya la venta de un artículo, deberá decrementarse el stock por la cantidad vendida de ese artículo.
- Si el valor llega a cero, abortar la transacción. Si hay menos de 3, emitir un mensaje.
- Dada una fecha, o una fecha de inicio y fecha de fin, regresar la cantidad total que se vendió en esa fecha/periodo.
- Permitir obtener el nombre de aquellos productos de los cuales hay menos de 3 en stock.
- De manera automática se genere una vista que contenga información necesaria para asemejarse a una factura de una compra.
- Crear al menos, un índice, del tipo que se prefiera y donde se prefiera. Justificar el porque de la elección en ambos aspectos.
- El número de venta debe tener un formato similar a "VENT-001", prefijo VENT, seguido de un guion y un número secuencial.
- Donde este presente el atributo domicilio, esta compuesto por estado, código postal, colonia, calle y número.
- El diseño debe satisfacer todos los principios de diseño, los requerimientos anteriores y un buen manejo de información.



Figure 2: Requerimientos

3.1 Análisis general

Se realizará un sistema de ventas para una papelería, para poder tener un registro de los productos que se almacenan, así como las ventas que se realicen.

El objetivo de la base de datos es dar seguimiento a datos tales como clientes, empleados, proveedores, inventario y productos con su respectiva categoría. Sin embargo, cabe mencionar que la conexión entre todos ellos, la encontramos centrada en las ventas de los productos. Por lo que también se consideró como una entidad. Los atributos y relaciones, se especificarán en el siguiente apartado.

3.2 Modelo Entidad-Relación.

Los elementos del modelo entidad – relación a identificar son: entidades, atributos, identificadores y relaciones.

- **Entidad:** Objeto exclusivo único en el mundo real que se está controlando. Se se quiere dar seguimiento.

Entidades identificadas:

- **PROVEEDOR:** Se desea almacenar información de aquel que provee productos.
- **PRODUCTO:** Se desea almacenar información del producto para venderlo al cliente y tener un registro en el inventario.
- **CATEGORÍA:** Se desea dividir en distintas categorías los respectivos productos
- **INVENTARIO:** Se desea almacenar información de los productos que nos permitan relacionarlos con las ventas, dado que tenemos que un precio de compra al proveedor y un precio de venta al cliente, esto con la finalidad de obtener la utilidad.
- **VENTA:** Se desea almacenar información de las ventas, de manera que se permita gestionar la salida de los productos y ofrecer un servicio óptimo al cliente.
- **CLIENTE:** Se desea almacenar información de aquella persona que utilice los servicios de la papelería, especialmente de la que lo hace regularmente.



Figure 3: Entidad

- **Atributos: Característica o rasgo de un tipo de entidad que describe la entidad.**

- PROVEEDOR:id, nombre, razón social, domicilio y teléfono
- PRODUCTO:id, marca, descripción, precio.
- CATEGORÍA:id, descripción.
- INVENTARIO: codigo de barras, precio de compra, fecha de compra, stock, precio de venta.
- VENTA: id, fecha, total
- CLIENTE:id, nombre, razón social, domicilio e email

- **Identificadores: son atributos que identifican las instancias de una entidad.**

- PROVEEDOR:identificado por idProv
- PRODUCTO:identificado por idProd
- CATEGORÍA:identificado por idCategoria
- INVENTARIO: identificado por codBarras
- VENTA:identificado por idVenta
- CLIENTE:identificado por idCliente

- **Relaciones:** las entidades se pueden asociar con otras mediante relaciones. Existen tres tipos de relaciones: 1..1 (uno a uno), 1..m (uno a muchos) y m..m (muchos a muchos).

El sistema guardara informacion de las categorias que se vendan por ejemplo: lapiz, libreta, borrador, etc, de las categorias se almanece un idCategoria y su descripción.

Tambien se almacenara informacion de los proveedores que surtan la papeleria, en este caso seria del contacto(idProveedor, nombre, telefono).

De los productos se deseara almacenar (el id del producto, marca, precio,descripcion, stock). 1 categoria aparece en varios productos y 1 producto solo tiene 1 categoria. 1 proveedor surte varios productos y 1 producto es surtido por un solo proveedor.

En el inventario se almacenaran los productos con su respectivo id, pero así mismo, se le asignará un código de barras,que se asociará con las ventas. Tenemos un precio de compra al proveedor, pero al requerir una utilidad por producto, el precio de venta al cliente, sería más elevado. También al tener un registro de los productos almacenados, en el atributo stock tendremos un control sobre los que aún se encuentran en bodega, y así, cuando los productos estén por terminarse, se comprarán más al proveedor.

La papeleria cuenta con un sistema de ventas. Se debera llevar tambien un registro de las ventas que se realicen en la papeleria , por cada venta se almacenara un idVenta, fecha y el total de la venta. Una venta tiene muchos productos y 1 producto puede estar en varias ventas.

De los clientes que hagan una venta se tendrá un idCliente, nombre, apellido paterno, apellido materno, direccion e email(pueden ser varios, al menos uno).

Un cliente realiza muchas ventas, 1 venta solo es realizada por 1 cliente.

Tambien se cuenta con usuarios(empleados) que seran los que se encargaran de realizar ventas y crear apartados, de ellos se almacena un idEmpleado, nombreUsuario y contraseña. 1 empleado puede realizar muchas ventas, pero una venta es realizada por un solo empleado, tambien un empleado crea varios apartados pero 1 apartado es creado por un solo usuario.

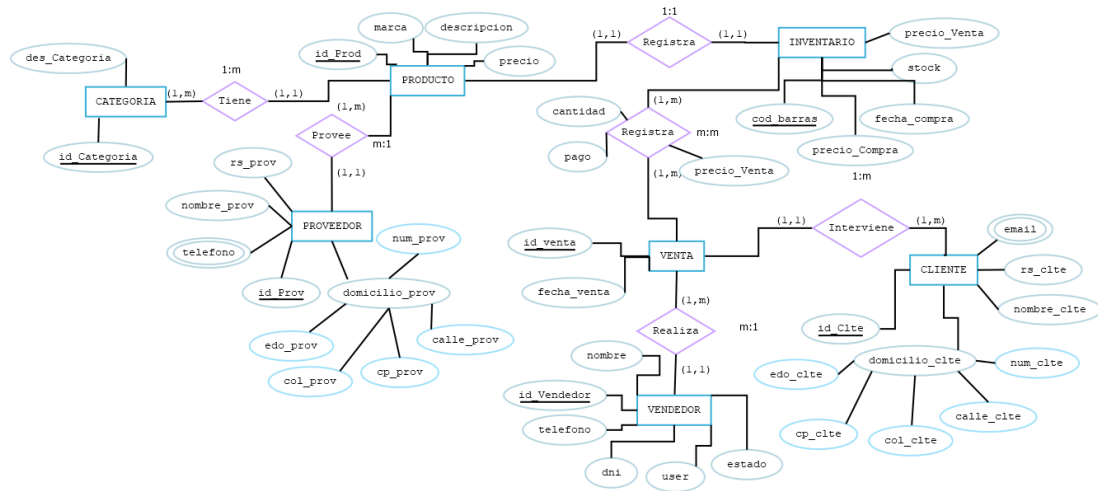
Modelo Entidad Relación final

Figure 4: Modelo Entidad Relación

3.3 Modelo Relacional

Modelo de organización y gestión de bases de datos consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos.

- **Mapeo Entidades:**

PROVEEDOR $id_{prov}(Pk), rs_{prov}, nom_{prov}, ap_{Patprov}, ap_{Matprov}, cp_{pov}, col_{prov}, calle_{prov} num_{prov}$

TELEFONO-PROVEEDOR $id_{prov}(PK, FK), tele_{fono}(PK)$

CLIENTE $id_{cliente}, rs_{cliente}, nom_{cliente}, edo_{cliente}, cp_{cliente}, col_{cliente}, calle_{cliente}, num_{cliente}$

EMAIL-CLIENTE $id_{cliente}(FK, PK), email_{CATEGORIA} id_{categoria}(PK), nom_{categoria}$

PRODUCTO $id_{prod}(PK), id_{prov}(FK), id_{categoria}(FK), marca, descripcion, precio$

INVENTARIO $cod_{barras}(PK), id_{prod}(FK), precio_{compra}, precio_{venta} fecha_{compra}, stock$

VENTA $id_{venta}(PK), id_{cliente}(FK), fecha_{venta}, cant_{art}, precio_{art}$

REGISTRAR $[cod_{barras}(FK), id_{venta}(FK)](PK), precio_{venta}, cantidad, pago, total$

VENDEDOR $id_{vendedor}(PK), nombre, DNI, user, estado$

TELEFONO-VENDEDOR $id_{vend}(PK, FK), telefono(PK)$

Modelo Relacional

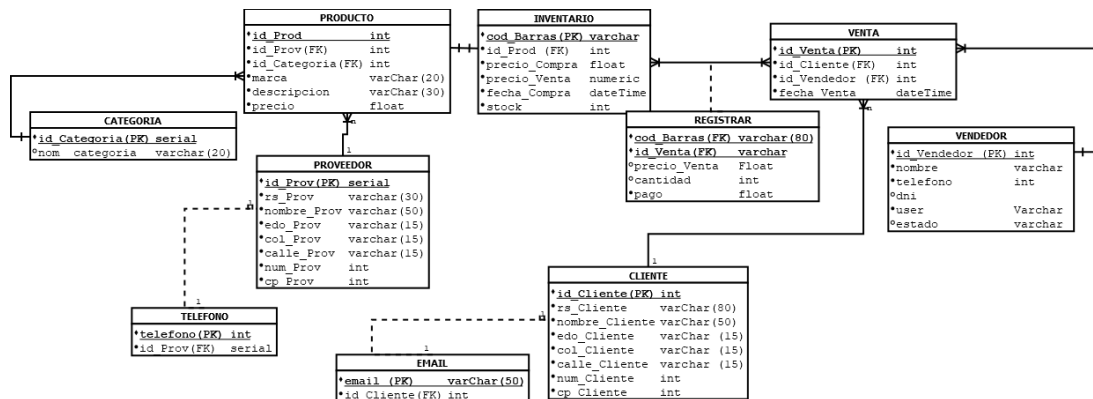


Figure 5: Modelo Relacional

3.4 Normalización

CUMPLEN 1FN

PROVEEDOR $id_{prov}(Pk), rs_{prov}, nom_{prov}, ap_{Patprov}, ap_{Matprov}, cppov, col_{prov}, calle_{prov}, num_{prov}$

TELEFONO-PROVEEDOR $id_{prov}(PK, FK), telefono(PK)$

CLIENTE $id_{cliente}, rs_{cliente}, nom_{cliente}, edo_{cliente}, cp_{cliente}, col_{cliente}, calle_{cliente}, num_{cliente}$

EMAIL-CLIENTE $id_{cliente}(FK, PK), email_{CATEGORIAid_{categoria}(PK)}, nom_{categoria}$

PRODUCTO $id_{prod}(PK), id_{prov}(FK), id_{categoria}(FK), marca, descripcion, precio$

INVENTARIO $cod_{barras}(PK), id_{prod}(FK), precio_{compra}, precio_{venta}, fecha_{compra}, stock$

VENTA $id_{venta}(PK), id_{cliente}(FK), fecha_{venta}, cant_{art}, precio_{art}$

REGISTRAR $[cod_{barras}(FK), id_{venta}(FK)](PK), precio_{venta}, cantidad, pago, total$

VENDEDOR $id_{vendedor}(PK), nombre, DNI, user, estado$

TELEFONO-VENDEDOR $id_{vend}(PK, FK), telefono(PK)$

CUMPLEN 2FN

PROVEEDOR $id_{prov}(Pk), rs_{prov}, nom_{prov}, ap_{Patprov}, ap_{Matprov}, cppov, col_{prov}, calle_{prov}, num_{prov}$

TELEFONO-PROVEEDOR $id_{prov}(PK, FK), telefono(PK)$

CLIENTE $id_{cliente}, rs_{cliente}, nom_{cliente}, edo_{cliente}, cp_{cliente}, col_{cliente}, calle_{cliente}, num_{cliente}$

EMAIL-CLIENTE $id_{cliente}(FK, PK), email_{CATEGORIAid_{categoria}(PK)}, nom_{categoria}$

PRODUCTO $id_{prod}(PK), id_{prov}(FK), id_{categoria}(FK), marca, descripcion, precio$

INVENTARIO $cod_{barras}(PK), id_{prod}(FK), precio_{compra}, precio_{venta}, fecha_{compra}, stock$

VENTA $id_{venta}(PK), id_{cliente}(FK), fecha_{venta}, cant_{art}, precio_{art}$

REGISTRAR $[cod_{barras}(FK), id_{venta}(FK)](PK), precio_{venta}, cantidad, pago, total$

VENDEDOR $id_{endedor}(PK), nombre, DNI, user, estado$

TELEFONO-VENDEDOR $id_{vend}(PK, FK), telefono(PK)$

No cumple 2FN

REGISTRAR $[cod_{barras}(FK), id_{venta}(FK)](PK), precio_{venta}, cantidad, pago, total$

Normalizando

REGISTRAR $[cod_{barras}(FK), id_{venta}(FK)](PK), precio_{venta}, cantidad, pago$

Agregamos atributos para facturación automática

REGISTRAR-PAGO $id_{venta}(PK), subtotal, IVA, total$

4 Implementación

4.1 Base de datos

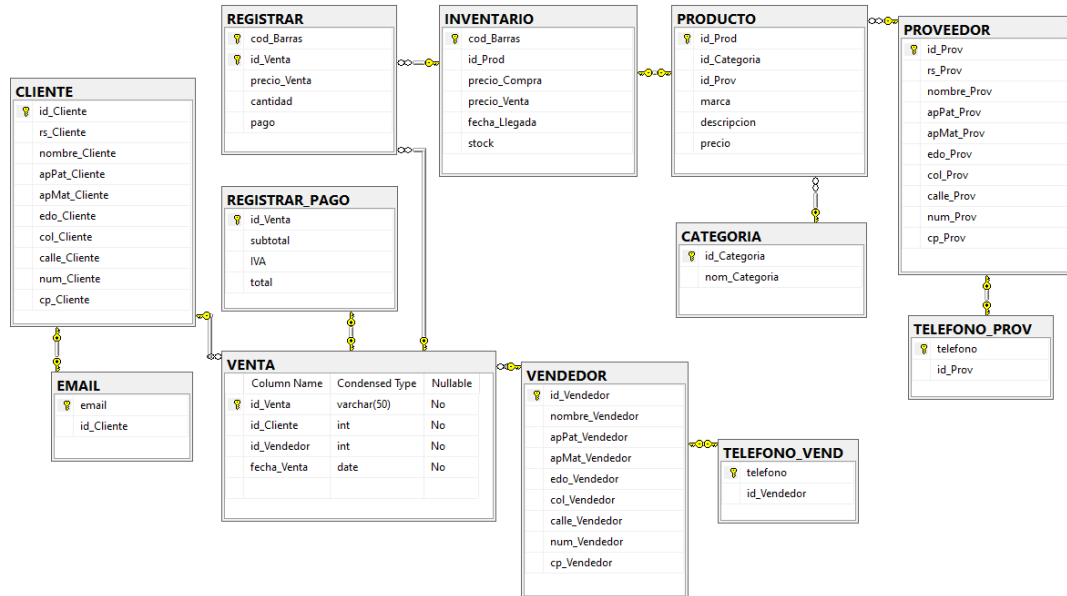


Figure 6: Modelo Relacional

4.2 Creación de tablas

Listing 1: DDL.Estructura de base de datos Papelería.

```

1
2  /*ALMACENAMIENTO DE DATOS DE PERSONAS (PROVEEDORES, CLIENTES, VENEDORES) */
3
4  CREATE TABLE PROVEEDOR(
5      id_Prov serial ,
6      rs_Prov varchar (80) not null ,
7      nombre_Prov varchar (30) not null ,
8      apPat_Prov varchar (30) not null ,
9      apMat_Prov varchar (30) null ,
10     edo_Prov varchar (30) not null ,
11     col_Prov varchar(30) not null ,
12     calle_Prov varchar(30) not null ,
13     num_Prov varchar(30) not null ,
14     cp_Prov int not null ,
15     CONSTRAINT "PK_PROVEEDOR" PRIMARY KEY (id_Prov)
16 );
17 /*Se registra el tel fono de proveedores en una nueva tabla ,
18 con la finalidad de evitar campos multivaluados , dado que se
19 puede almacenar m s de un campo*/
20
21 CREATE TABLE TELEFONO_PROV(
22     telefono int not null ,
23     id_Prov serial ,
24     UNIQUE (id_prov),
25     CONSTRAINT "PK_TELEFONO_PROV" PRIMARY KEY (telefono),
26     CONSTRAINT "FK_TELEFONO_PROVEEDOR" FOREIGN KEY (id_Prov)
27     REFERENCES PROVEEDOR(id_Prov)
28 );
29
30
31 CREATE TABLE CLIENTE(
32     id_Cliente serial ,
33     rs_Cliente varchar (80) not null ,
34     nombre_Cliente varchar (30) not null ,
35     apPat_Cliente varchar (30) not null ,
36     apMat_Cliente varchar (30) null ,
37     edo_Cliente varchar (30) not null ,
38     col_Cliente varchar(30) not null ,
39     calle_Cliente varchar (30) not null ,
40     num_Cliente varchar(30) not null ,
41     cp_Cliente int not null ,
42     CONSTRAINT "PK_CLIENTE" PRIMARY KEY (id_cliente)
43 );
44
45 /*Se registra el email de clientes en una nueva tabla ,
46 con la finalidad de evitar campos multivaluados (Se almacena al menos
47 uno, con la restricci n de no aceptar registros nulos , pero pueden
48 ser m s.*/
49 CREATE TABLE EMAIL(

```

```
50     email varchar(50) not null ,
51     id_Cliente serial ,
52     UNIQUE (id_Cliente),
53     CONSTRAINT "PK_EMAIL" PRIMARY KEY (email),
54     CONSTRAINT "FK_EMAIL_CLIENTE" FOREIGN KEY (id_Cliente)
55     REFERENCES CLIENTE(id_Cliente)
56 );
57
58
59
60 CREATE TABLE VENDEDOR(
61     id_Vendedor serial ,
62     nombre_Vendedor varchar (30) not null ,
63     apPat_Vendedor varchar (30) not null ,
64     apMat_Vendedor varchar (30) null ,
65     edo_Vendedor varchar (30) not null ,
66     col_Vendedor varchar(30) not null ,
67     calle_Vendedor varchar (30) not null ,
68     num_Vendedor varchar(30) not null ,
69     cp_Vendedor int not null ,
70     CONSTRAINT "PK_VENDEDOR" PRIMARY KEY (id_Vendedor)
71 );
72
73 /*Se registra el tel fono de proveedores en una nueva tabla ,
74 con la finalidad de evitar campos multivaluados , dado que se
75 puede almacenar m s de un campo*/
76
77
78 CREATE TABLE TELEFONO_VEND(
79     telefono int not null ,
80     id_Vendedor serial ,
81     UNIQUE (id_Vendedor ) ,
82     CONSTRAINT "PK_TELEFONO_VEN" PRIMARY KEY (telefono),
83     CONSTRAINT "FK_TELEFONO_VENDEDOR" FOREIGN KEY (id_Vendedor )
84     REFERENCES VENDEDOR(id_Vendedor)
85 );
86
87
88
89 /*ALMACENAMIENTO DE INFORMACIN RELACIONADA A LOS PRODUCTOS*/
90
91
92 —Creacion de indice
93 —CREATE INDEX Indice_Categoria on CATEGORIA(id_Categoria);
94
95 CREATE TABLE CATEGORIA (
96     id_Categoria serial ,
97     nom_Categoria varchar (30),
98     CONSTRAINT "PK_CATEGORIA" PRIMARY KEY (id_categoria)
99 );
100
101 CREATE TABLE PRODUCTO(
```

```

102     id_Prod serial ,
103     id_Categoria int not null,—serial ,
104     id_Prov int not null,—serial ,
105     marca varchar (60),
106     descripcion varchar (30),
107     precio numeric not null ,
108     CONSTRAINT "PK_PRODUCTO" PRIMARY KEY (id_Prod),
109     CONSTRAINT "FK_PRODUCTO_CATEGORIA" FOREIGN KEY (id_Categoria)
110     REFERENCES CATEGORIA(id_Categoria),
111     CONSTRAINT "CK_PRECIO_PRODUCTO" CHECK (precio >0),
112     CONSTRAINT "FK_PRODUCTO_PROVEEDOR" FOREIGN KEY (id_Prov)
113     REFERENCES PROVEEDOR(id_Prov)
114 );
115
116 /*Almacena informaci n de los productos , relacionada a la compra al proveedor.
117
118 As mismo se asigna un codigo de barras por producto y el precio de venta al cliente .
119
120 Por otra parte tenemos el atributo stock que nos permite tener un control sobre
121 la cantidad de productos que hay almacenados , de esta forma , es vol til y cambia
122 con respecto a las ventas realizadas.*/
123
124 CREATE TABLE INVENTARIO(
125     cod_Barras varchar(80) not null ,
126     id_Prod serial ,
127     precio_Compra numeric not null ,
128     precio_Venta numeric not null ,
129     fecha_Llegada date not null ,
130     stock int not null ,
131     UNIQUE (id_Prod),
132     CONSTRAINT "PK_INVENTARIO" PRIMARY KEY (cod_Barras) ,
133     —CONSTRAINT "CK_STOCK" CHECK (stock >3),
134     CONSTRAINT "CK_PRECIO" CHECK (precio_Venta >0),
135     CONSTRAINT "FK_PRODUCTO_INVENTARIO" FOREIGN KEY (id_Prod)
136     REFERENCES PRODUCTO(id_Prod)
137 );
138
139
140
141
142
143 /*ALMACENAMIENTO DE INFORMACIN RELACIONADA A LAS VENTAS DE LOS PRODCUTOS*/
144
145
146 CREATE TABLE VENTA(
147     id_Venta varchar(50),
148     id_Cliente serial ,
149     id_Vendedor int not null ,
150     fecha_Venta date not null ,
151     CONSTRAINT "PK_VENTA" PRIMARY KEY (id_Venta),
152     CONSTRAINT "FK_VENTA_CLIENTE" FOREIGN KEY (id_Cliente)
153     REFERENCES CLIENTE(id_Cliente),

```



```
154     CONSTRAINT "FK_VENTA_VENDEDOR" FOREIGN KEY (id_Vendedor)
155     REFERENCES VENDEDOR(id_Vendedor)
156 );
157
158 /*La tabla registrar , tiene una llave primaria compuesta , esto con la
159 finalidad de almacenar m s de 1 producto en 1 venta .
160 Con la estructura de la siguiente tabla , podemos almacenar la cantidad de
161 productos del mismo tipo , para tener el total de pago por producto.*/
162
163 CREATE TABLE REGISTRAR(
164     cod_Barras varchar(80) not null ,
165     id_Venta varchar(50) not null ,
166     precio_Venta numeric not null ,
167     cantidad int not null ,
168     pago numeric not null ,
169     CONSTRAINT "PK_REGISTRAR" PRIMARY KEY (id_Venta , cod_Barras),
170     CONSTRAINT "FK_REGISTRAR_INVENTARIO" FOREIGN KEY(cod_Barras)
171     REFERENCES INVENTARIO (cod_Barras),
172     CONSTRAINT "FK_REGISTRAR_VENTA" FOREIGN KEY(id_Venta)
173     REFERENCES VENTA (id_Venta)
174
175 );
176
177 /*La tabla registrar_pago , nos permite almacenar informaci n del pago por venta realiza
178
179
180 CREATE TABLE REGISTRAR_PAGO(
181     id_Venta varchar(50) not null ,
182     subtotal numeric not null ,
183     IVA numeric not null ,
184     total numeric not null ,
185     CONSTRAINT "PK_REGISTRAR_PAGO" PRIMARY KEY (id_Venta),
186     CONSTRAINT "FK_REGISTRAR_PAGO_VENTA" FOREIGN KEY(id_Venta)
187     REFERENCES VENTA (id_Venta)
188 );
```

4.3 Agregado de información

CLIENTE: Información de aquella persona que utilice los servicios de la papelería.









							
	id_cliente [PK] integer	rs_cliente character varying (80)	nombre_cliente character varying (30)	apmat_cliente character varying (30)	apmat_cliente character varying (30)	edo_cliente character varying (30)	co_cliente character varying (30)
1	1	WageWorks, Inc.	Randolph	Lenard	Lease	California	Welch
2	2	Tanger Factory Outlet Center...	Damon	Peres	Gauvain	Indiana	Stoughton
3	3	Graphic Packaging Holding ...	Clemente	Burie	Cockran	Delaware	Tailsman
4	4	Acacia Research Corporation	Sutton	Milroy	Gladhill	Arizona	Amoth
5	5	Evogene Ltd.	Norean	Stack	Di Matteo	Texas	Kim
6	6	Qiagen N.V.	Artus	Cutting	Scullard	North Carolina	Oriole
7	7	ADMA Biologics Inc	Simona	Cohen	Monini	Texas	Graedel
8	8	First Trust Emerging Market...	Florian	Cawthery	MacDaid	Mississippi	Brentwood
9	9	Astrotech Corporation	Ced	Craxford	Plewright	District of Columbia	Starling
10	10	Oxbridge Re Holdings Limited	Lee	Vanichkov	Rooney	Connecticut	Ruskin
11	11	American Airlines Group, Inc.	Chanda	Lamcken	Aust	Hawaii	David
12	12	TICC Capital Corp.	Davine	Chambers	Carrel	Virginia	Monterey
13	13	The Goodyear Tire & Rubber ...	Orella	Fahy	Addyman	Louisiana	Merchant
14	14	Diamond Offshore Drilling, In...	Bucky	Kendell	Baber	Florida	Birchwood
15	15	First Trust Mega Cap AlphaD...	Gene	Gillet	Higbin	New York	Artisan
16	16	Cementos Pacasmayo S.A.A.	Georgina	Phizaclea	Davern	Indiana	Anniversary
17	17	Belden Inc	Vinni	Kewish	Sacco	Indiana	Leroy
18	18	Targa Resources Partners LP	Rolland	Mabb	Livoir	Arizona	Coleman
19	19	Citigroup Inc.	Massimiliano	Allawy	Fetters	New York	Quincy
20	20	USA Truck, Inc.	Jilly	Brumby	Saw	Illinois	Kropf

Figure 7: Tabla Cliente

PROVEEDOR: Se desea almacenar información de aquel que provee productos.









							
	id_prov [PK] integer	rs_prov character varying (80)	nombre_prov character varying (30)	apmat_prov character varying (30)	apmat_prov character varying (30)	edo_prov character varying (20)	co_prov character varying (20)
1	1	Nuveen New York Municipal ...	Caye	Boulthwood	Duckhouse	Florida	Haas
2	2	Community First Bancshare...	Shoshanna	Kitcat	Wiseman	New Mexico	Park Meadow
3	3	Deutsch Bk Contingent Cap ...	Bathsheba	Matyushonok	Juniper	New York	5th
4	4	Consolidated Communicatio...	Karlie	Smalley	Prinnett	Florida	Arizona
5	5	CDW Corporation	Ulrick	Earsman	McGrah	Nebraska	Ridgeway
6	6	Provident Financial Holdings...	Kylie	Enion	Vedntyev	Texas	Myrtle
7	7	Ares Dynamic Credit Allocati...	Preston	Edmed	O Cuddie	Kentucky	Fuller
8	8	INTL FCStone Inc.	Rici	Hawarden	Darcey	Michigan	Fair Oaks
9	9	Global Indemnity Limited	Nyssa	Haddrell	Bartles	Iowa	Mockingbird
10	10	Hilton Grand Vacations Inc.	Nada	Allardyce	Bisseker	California	Sullivan
11	11	iFresh Inc.	Agna	Stichall	Hartford	West Virginia	Portage
12	12	Coca-Cola Company (The)	Les	Olirenshaw	Connelly	Massachusetts	Moland
13	13	Eagle Pharmaceuticals, Inc.	Reinaldo	Normansell	Feldmus	Wisconsin	Carpenter
14	14	Nuance Communications, Inc.	Mignon	Loseby	Mabson	New York	Meadow Valley
15	15	J. Alexanders Holdings, Inc.	Marshal	Cobain	Mcmanaman	California	Westport
16	16	Hamilton Lane Incorporated	Gib	Heinzler	Jerome	Kansas	Hollow Ridge
17	17	First Trust Nasdaq Pharmac...	Susannah	Santer	Coetzee	Illinois	Petterle
18	18	Calamos Dynamic Convertib...	Rosemonde	Loseke	Shelbourne	Alabama	Lindbergh
19	19	Cohen & Steers Reit and Pref...	Lefty	Beraneck	Makepeace	Wisconsin	Shopko
20	20	Vishay Precision Group, Inc.	Minny	Enterlie	Duffrie	Hawaii	Schiller

Figure 8: Tabla Proveedor

PRODUCTO: Se desea almacenar información del producto para venderlo al cliente y tener un registro en el inventario.

	id_prod [PK] integer	id_categoria integer	id_prov integer	marca character varying (60)	descripcion character varying (30)	precio numeric
1	1	1	1	Tenaris S.A.	Oso de peluche	61.88
2	2	2	2	Arconic Inc.	Impresión de carteles	147.97
3	3	3	3	China Distance Education H...	Recarga 150	207.82
4	4	4	4	Omega Flex, Inc.	Folder pasta dura	73.33
5	5	5	5	Macys Inc	Paquete de 1000 hojas	153.2
6	6	6	6	Hennessy Advisors, Inc.	Reloj digital	113.59
7	7	7	7	Green Plains, Inc.	Audifonos	217.49
8	8	8	8	Invesco Bond Fund	Lapicera de colores	283.16
9	9	9	9	Tunlu Corporation	Hilo elastico	128.92
10	10	10	10	Landcadia Holdings, Inc.	USB de 100 gb	161.17
11	11	1	11	Tunlu Corporation	Globos	191.97
12	12	2	12	Invesco High Income Trust II	Impresión de playeras	202.08
13	13	3	13	argenx SE	Recarga de 150 Telcel	153.5
14	14	4	14	Liberty Broadband Corporati...	Engrapadora	209.37
15	15	5	15	Lumentum Holdings Inc.	Paquete hoja A4	113.89
16	16	6	16	Nanometrics Incorporated	Audifonos economicos	80.59
17	17	7	17	Nuveen Select Tax Free Inco...	Alcohol Gel	180.14
18	18	8	18	Alphatec Holdings, Inc.	Carpeta universitaria	181.38
19	19	9	19	Great Elm Capital Corp.	Hilo decoracion	62.64
20	20	10	20	BGC Partners, Inc.	USB 16 GB	87.59
21	21	1	1	KalVista Pharmaceuticals, In...	Chocolates personalizados	214.24
22	22	8	8	H&E Equipment Services, Inc.	Cuadernos sin espiral	233.28
23	23	9	9	Caladrius Biosciences, Inc.	Kit para dobladillos	222.09
24	24	10	10	Post Holdings, Inc.	Contenedor 1 lt	151.33

Figure 9: Tabla Producto

CATEGORÍA: Se desea dividir en distintas categorías los respectivos productos

	id_categoria [PK] integer	nom_categoria character varying (30)
1	1	Regalo
2	2	Impresión
3	3	Recarga
4	4	Oficina
5	5	Papel
6	6	Tecnología
7	7	Accesorios
8	8	Escolar
9	9	Mercería
10	10	Almacenamiento

Figure 10: Tabla Categoría

INVENTARIO: Se desea almacenar información de los productos que nos permitan relacionarlos con las ventas, dado que tenemos que un precio de compra al proveedor y un precio de venta al cliente, esto con la finalidad de obtener la utilidad.

	cod_barras [PK] character varying (80)	id_prod integer	precio_compra numeric	precio_venta numeric	fecha_llegada date	stock integer
1	SK54 3772 2355 7482 3156 1717	1	61.88	71.15	2019-11-05	48
2	LV88 TOYX DSO0 IHXZ 8LK0 0	2	147.97	170.16	2019-03-06	77
3	SK57 2720 0365 1857 8838 3618	3	207.82	238.993	2019-05-09	66
4	IE18 OVAS 5622 0407 2366 34	4	73.33	84.32	2019-05-23	18
5	LI54 8712 82PZ 0GKL JUHZ X	5	153.2	176.18	2019-05-21	51
6	CZ29 3049 8021 4441 1380 1105	6	113.59	130.62	2019-06-05	49
7	VG07 TNPB 3067 9679 3791 4978	7	217.49	250.11	2019-03-30	26
8	AL47 2753 8452 2UHT 2N9T XYY...	8	283.16	325.63	2019-06-14	4
9	KZ95 931N JRGO 571H ZOYK	9	128.92	148.25	2019-05-12	53
10	BH94 ZAHC LVP5 6040 WSX9 NH	10	161.17	200.12	2019-04-29	90
11	FR15 0426 2996 71UI PXE1 T1SE ...	11	191.97	250.97	2019-02-10	67
12	FR09 2763 8619 10NZ ZAO1 Y3KL...	12	202.08	252.08	2019-01-01	93
13	MD02 P997 E0EW ID2V TARL OS...	13	153.5	210.5	2019-06-02	89
14	SA32 571F DFRW XYQU ONLJ GU...	14	209.37	258.37	2019-05-03	60
15	MK74 404S FVVH FAN3 J03	15	113.89	195.89	2019-11-05	40
16	FR11 5418 3953 83NK J50J ZAS...	16	80.59	130.59	2019-03-06	12
17	GT28 NYRO ZV70 3QFV CTFT 3A...	17	180.14	210.14	2019-05-09	53
18	KW63 OAKH F1ND VW3L FASS VI...	18	181.38	236.38	2019-05-23	99
19	GE67 HV16 6467 3257 6999 19	19	62.64	150.64	2019-05-21	91
20	SK57 2720 0365 1857 8838 3619	20	87.59	130.59	2019-06-05	96
21	IE18 OVAS 5622 0407 2366 33	21	214.24	300	2019-03-30	43
22	LI54 8712 82PZ 0GKL JUHZ Y	22	172.88	250	2019-06-14	23
23	CZ29 3049 8021 4441 1380 1115	23	64.63	100	2019-05-12	73
24	VG07 TNPB 3067 9679 3791 49J8	24	165.95	200	2019-04-29	55

Figure 11: Tabla Inventario

VENTA: Se desea almacenar información de las ventas, de manera que se permita gestionar la salida de los productos, en relación al cliente.

	id_venta [PK] character varying (50)	id_cliente integer	fecha_venta date
1	VENT-001		2015-05-24
2	VENT-002		2019-12-15
3	VENT-003		2019-08-30
4	VENT-004		2017-06-11
5	VENT-005		2015-08-07
6	VENT-006		2016-11-25
7	VENT-007		2017-11-04
8	VENT-008		2018-06-16
9	VENT-009		2016-09-27
10	VENT-010		2017-05-12
11	VENT-011		2020-04-02
12	VENT-012		2019-09-23
13	VENT-013		2018-03-31
14	VENT-014		2016-12-09
15	VENT-015		2018-12-27
16	VENT-016		2015-08-26
17	VENT-017		2019-10-22
18	VENT-018		2018-01-30
19	VENT-019		2019-09-02
20	VENT-020		2018-07-19
21	VENT-021		2019-05-05
22	VENT-022		2017-01-24
23	VENT-023		2016-10-18

Figure 12: Tabla Venta

REGISTRAR: Se desea almacenar información de las ventas, de manera que se permita gestionar la salida de los productos, en relación al inventario.

	cod_barras [PK] character varying (80)	id_venta [PK] character varying (50)	precio_venta numeric	cantidad integer	pago numeric
1	SK54 3772 2355 7482 3156 1717	VENT-001	71.15	1	71.15
2	LV88 TOYX DS00 IHXZ 8LK0 0	VENT-001	170.16	7	1191.12
3	SK57 2720 0365 1857 8838 3618	VENT-002	238.993	2	477.986
4	IE18 OVAS 5622 0407 2366 34	VENT-002	84.32	7	590.24
5	LI54 8712 8ZPZ 0GKL JUHZ X	VENT-002	176.18	3	528.54
6	CZ29 3049 8021 4441 1380 1105	VENT-003	130.62	6	783.72
7	VG07 TNPB 3067 9679 3791 4978	VENT-003	250.11	8	2000.88
8	AL47 2753 8452 2UHT 2N9T XYY...	VENT-004	325.63	10	3256.3
9	KZ95 931N JRGO 571H ZQYK	VENT-005	148.25	2	296.5
10	BH94 ZAHC LVP5 6O4O WSX9 NH	VENT-005	200	1	200
11	FR15 0426 2996 71UI PXE1 T1SE ...	VENT-006	250.97	5	1254.85
12	FR09 2763 8619 10NZ ZAO1 Y3KI...	VENT-007	252.08	5	1260.4
13	MD02 P997 E0EW ID2V TARL OS...	VENT-008	210.5	3	631.5
14	SA32 57IF DFRW XYQU ONLJ GU...	VENT-009	258.37	4	1033.48
15	MK74 404S FVVH FAN3 J03	VENT-010	195.89	1	195.89
16	FR11 5418 3953 83NK JSOJ ZAS...	VENT-010	130.59	3	391.77
17	GT28 NYRO ZVT0 3QFV CTFT 3A...	VENT-011	210.14	8	1681.12
18	KW63 OAKH F1ND VW3L FASS VI...	VENT-012	236.38	3	709.14
19	GE67 HV16 6467 3257 6999 19	VENT-013	150.64	2	301.28
20	SK57 2720 0365 1857 8838 3618	VENT-014	130.59	4	522.36
21	IE18 OVAS 5622 0407 2366 34	VENT-015	250.24	4	1000.96
22	LI54 8712 8ZPZ 0GKL JUHZ X	VENT-015	200.88	1	200.88
23	CZ29 3049 8021 4441 1380 1105	VENT-016	130.62	2	201.26
24	VG07 TNPB 3067 9679 3791 4978	VENT-017	250.11	1	139.95

Figure 13: Tabla Registrar

4.4 Consultas

Listing 2: Consulta periodo fecha por producto

```

1  /*Dada una fecha de inicio y fin, numero de productos vendidos,
2  del m s vendido al menos. Organizados por producto.*/
3
4  CREATE OR REPLACE FUNCTION venta_periodo(v varchar, v varchar)
5  returns TABLE (cod_Barras varchar(80), cantidad bigint)
6  as
7  $$
8      declare fin date;
9      declare inicio date;
10     BEGIN
11         fin = date($2);
12         inicio = date($1);
13         raise notice 'Fecha inicio: %', inicio;
14         raise notice 'Fecha fin: %', fin;
15     RETURN QUERY SELECT r.cod_barras, sum(r.cantidad)
16                     FROM registrar r
17                     inner join VENTA V ON R.id_Venta = V.id_Venta
18                     GROUP BY R.COD_BARRAS, R.CANTIDAD, V.ID_VENTA
19                     HAVING fecha_Venta >= inicio and fecha_Venta <= fin
20                     ORDER BY R.cantidad DESC;
21     END;
22     $$
23     LANGUAGE PLPGSQL;
24
25     -----REALIZANDO LA CONSULTA -----
26
27     SELECT VENTA_PERIODO( '2015-05-24', '2018-05-24' );

```

	venta_periodo record
1	(*AL47 2753 8452 2UHT 2N9T XYYS HB7N",10)
2	(*LV88 TOYX DSO0 IHXZ 8LK0 0",7)
3	(*FR09 2763 8619 10NZ ZAO1 Y3KI D85",5)
4	(*FR15 0426 2996 71UI PXE1 T1SE 086",5)
5	(*SA32 571F DFRW XYQU ONLJ GUMS",4)
6	(*SK57 2720 0365 1857 8838 3618",4)
7	(*FR11 5418 3953 83NK J50J ZASG 563",3)
8	(*MD02 P997 E0EW ID2V TARL OSQG",3)
9	(*CZ29 3049 8021 4441 1380 1105",2)
10	(*GE67 HV16 6467 3257 6999 19",2)
11	(*KZ95 931N JRG0 571H ZQYK",2)
12	(*AL47 2753 8452 2UHT 2N9T XYYS HB7N",1)
13	(*BH94 ZAHC LVP5 6040 WSX9 NH",1)
14	(*LI54 8712 8ZPZ 0GKL JUHZ X",1)
15	(*MK74 404S FVVH FAN3 J03",1)
16	(*SK54 3772 2355 7482 3156 1717",1)

Figure 14: Ventas por producto,dado un periodo de fecha

Listing 3: Consulta periodo fecha de productos totales vendidos

```

1  /* Venta periodo , dada una fecha de inicio y fin , total productos */
2
3  CREATE OR REPLACE FUNCTION TOTALventa_periodo( varchar , varchar )
4  returns int
5  as
6  $$
7      declare fin date;
8      declare inicio date;
9      declare totalProductos int;
10     BEGIN
11         fin = date($2);
12         inicio = date($1);
13         raise notice 'Fecha inicio: %', inicio;
14         raise notice 'Fecha fin: %', fin;
15         totalProductos:= (SELECT SUM(R.cantidad) as CantidadProductosVendidos
16                           FROM Venta V
17                           INNER JOIN Registrar R ON R.id_Venta=v.id_Venta
18                           WHERE fecha_Venta between inicio and fin);
19         RETURN totalProductos;
20     END;
21     $$
22     LANGUAGE PLPGSQL;
23
24     -----REALIZANDO LA CONSULTA-----
25     SELECT totalVENTA_PERIODO( '2015-05-24' , '2018-05-24' );

```

totalventa_periodo	
integer	
1	52

Figure 15: Ventas totales,dado un periodo de fecha

Listing 4: Función que regresa la utilidad dado un código de barras

```

1  /* Funci n que regresa la utilidad dado un c digo de barras */
2  CREATE OR REPLACE FUNCTION UTILIDAD ( varchar(80))
3  RETURNS numeric
4  AS $$
5      declare compra numeric;
6      declare venta numeric;
7      declare utilidad numeric;
8      BEGIN
9      compra:= (SELECT precio_Compra FROM INVENTARIO i WHERE i.cod_Barras = $1);
10     venta := (Select precio_venta FROM INVENTARIO i where i.cod_Barras = $1);

```

```

11     utilidad = venta - compra;
12     RETURN utilidad;
13     END;
14 $$LANGUAGE PLPGSQL;
15
16
17 -----EJECUTANDO LA FUNCION UTILIDAD-----
18 SELECT *FROM utilidad ('GT28 NYRO ZVT0 3QFV CTFT 3AKV RV8Q');

```

	utilidad
	numeric
1	30.00

Figure 16: Utilidad dado un codigo de barras

Listing 5: Factura

```

1
2  /*-----Vista de una factura-----*/
3  CREATE VIEW FACTURA
4  AS
5  Select* FROM(
6      SELECT 1 AS Factura , 'Datos de Factura:' as PapeleriaBaseDeDatos ,
7      CONCAT('Cliente: ', CAST(C.id_Cliente AS varchar(10))) as Tel55070220 ,
8      CONCAT('Fecha: ',CAST(V.fecha_Venta AS varchar(10))) as AV20DENOVIEMBRE,
9      CONCAT('Venta: ', v.id_Venta) as Num1024, ' ' as Pago
10     FROM CLIENTE C
11     INNER JOIN Venta V ON V.id_Cliente = C.id_Cliente
12     Where v.id_Venta= 'VENT-001'
13  UNION
14     SELECT 2,CONCAT( 'Facturar A: ',C.appat_Cliente , ' ',C.nombre_Cliente ),
15     C.rs_Cliente ,CONCAT(C.calle_Cliente , ' #',C.num_Cliente , ' ',
16     Col. ',C.col_Cliente , ' ', CP. ',C.cp_Cliente),
17     E.email as CorreoElectronico , ' '
18     FROM CLIENTE C
19     INNER JOIN Venta V ON V.id_Cliente = C.id_Cliente
20     INNER JOIN email E ON E.id_Cliente = C.id_Cliente
21     Where v.id_Venta= 'VENT-001'
22  UNION
23     SELECT '3', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
24  UNION
25     SELECT '3', 'Codigo Barras', 'Producto', 'Precio unitario', 'Cantidad ', ' '
26  UNION
27     SELECT 4, r.cod_Barras , P.descripcion as Producto ,
28     CAST(r.precio_Venta AS varchar(10)),
29     CAST(r.cantidad AS varchar(10)),CAST(r.pago AS varchar(10))
30     From Venta v

```



```

31     inner join registrar r on v.id_Venta = r.id_Venta
32     inner join inventario I on r.cod_Barras= i.cod_Barras
33     inner join producto P on p.id_Prod=i.id_Prod
34     Where v.id_Venta= 'VENT-001'
35 UNION
36     SELECT 5, ' ', ' ', ' ', ' ', 'SUBTOTAL', CAST(SUM(r.pago) AS varchar(10))
37     FROM Venta V
38     inner join registrar r on v.id_Venta = r.id_Venta
39     Where v.id_Venta= 'VENT-001'
40 UNION
41     SELECT 6, ' ', ' ', ' ', ' ', 'IVA 16%', CAST((SUM(r.pago))*0.16 AS varchar(10))
42     FROM Venta V
43     inner join registrar r on v.id_Venta = r.id_Venta
44     Where v.id_Venta= 'VENT-001'
45 UNION
46     SELECT 7, ' ', ' ', ' ', ' ', 'TOTAL', CAST((SUM(r.pago)+(SUM(r.pago))*0.16) AS
47     varchar(10))
48     FROM Venta V
49     inner join registrar r on v.id_Venta = r.id_Venta
50     Where v.id_Venta= 'VENT-001'
51 ) as FACT
52 ORDER BY Factura;
53
54 -----EJECUCION DE FACTURA-----
55 Select * From FACTURA

```

	factura integer	papeleriabasededatos text	tel55070220 text	av20denoviembre text	num1024 text	pago text
1	1	Datos de Factura:	Ciliente: 1	Fecha: 2015-05-24	Venta: VENT-001	
2	2	Facturar A: Lenard Randolph	WageWorks, Inc.	Jackson #6516, Col.Welc...	acaizley0@lulu.com	
3	3					
4	3	Codigo Barras	Producto	Precio unitario	Cantidad	
5	4	LV88 TOYX DSO0 IHXZ 8LK0 0	Impresión de carteles	170.16	7	1191.12
6	4	SK54 3772 2355 7482 3156 ...	Oso de peluche	71.15	1	71.15
7	5				SUBTOTAL	1262.27
8	6				IVA 16%	201.9632
9	7				TOTAL	1464.2332

Figure 17: Factura de una venta

Listing 6: Índice

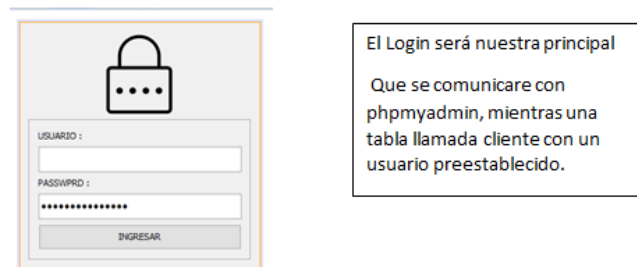
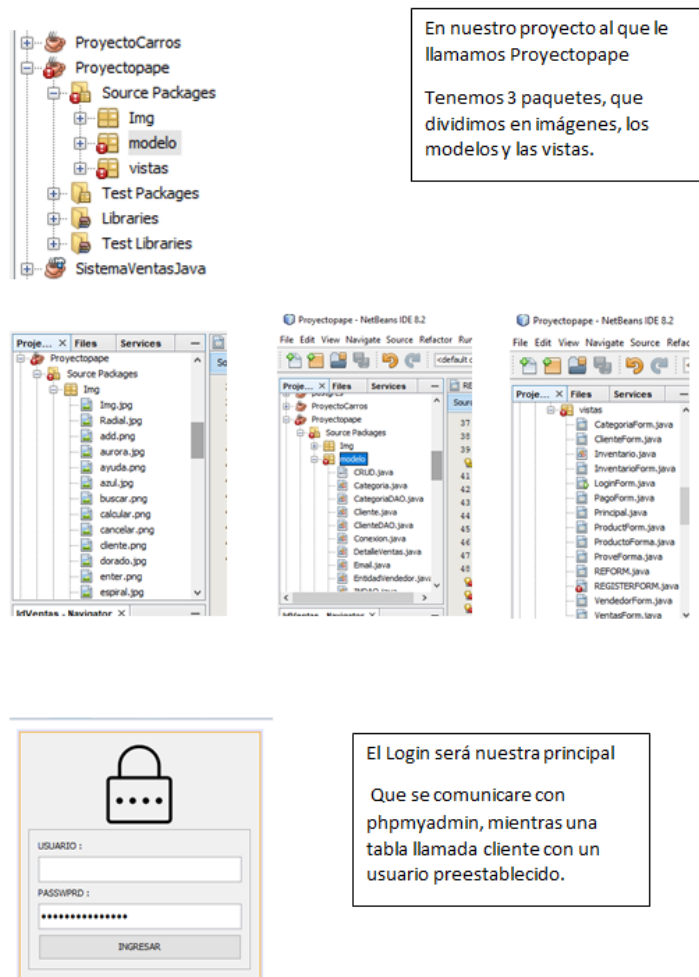
```
1
2 /* Creaci n de ndice */
3 —El ndice se crea en el c digo de barra , con la finalidad de realizar
4 una b squeda mas r pida al hacer ventas.
5
6 CREATE INDEX Indice_Baras on INVENTARIO(cod_Barras );
```

Listing 7: Verificaci3n de stock

```
1
2 /*—————CUANDO STOCK SEA MENOR A 3—————*/
3 CREATE OR REPLACE FUNCTION STOCK ()
4 RETURNS TABLE (cod_Barras varchar(80), cantidad int )
5 AS
6     $$
7     BEGIN
8         RETURN query
9             SELECT i.cod_Barras , i.stock
10            FROM INVENTARIO i
11            WHERE stock <= 3;
12     END;
13 $$LANGUAGE PLPGSQL;
14 —————EJECUCION STOCK—————
15 select *from stock ();
```

5 Presentación

Para la implementación de la interfaz, se identificó el conocimiento de los miembros del equipo acerca de la integración de la base de datos a un entorno gráfico, acorde a los requerimientos del proyecto. Del resultado de ese análisis se decidió desarrollar la interfaz por medio de Java AWT (Abstract Window Toolkit), integrándose con las bibliotecas de Java para manejar bases de datos con SQL. Posteriormente se determinaron las ventanas y los componentes interactivos necesarios (botones, cuadros de texto, listas, etc.) para que el usuario final intervenga y utilice el sistema de una manera adecuada, asegurando que se respeten las reglas e integridad de la información en la base de datos. Además se asegura que los datos sólo se modifiquen por un determinado usuario con contraseña. La interfaz gráfica también fue mejorada en el aspecto estético, colocando el logo y otros elementos gráficos.



Nuestro proyecto se comunicará con phpmyadmin con una interfaz en Java.

Figure 18: Interfaz

6 Conclusiones

Silva Barrera Brandon:

El proyecto tuvo como dificultad principal, que se reformó al utilizar los avances de dos equipos, por lo que se requirió una buena comunicación y un trabajo de análisis para lograr su integración. Gran parte de los componentes de ambos equipos tuvieron que ser modificados en algunas de sus partes o de forma total, lo que requirió una mayor inversión de tiempo y esfuerzo. El intercambio de ideas entre los nuevos integrantes del proyecto fue fundamental para desarrollar los requerimientos del proyecto con el trabajo ya hecho, por lo que es un acierto tomar en cuenta las opiniones de todos y resolver las diferencias de manera que se llegue a un consenso de cómo se resolverá el proyecto. Las herramientas colaborativas para que todos visualicen el avance del proyecto y lo modifiquen fueron esenciales para el progreso del proyecto, además de tener acceso a información por medio de internet, lo que ayudó a cumplir con el proyecto.

Garcia Cruz Diana Aide:

Para crear bases de datos sólidas es necesario tener los conocimientos de normalización y mapeo así como conceptualizar los requerimientos y el funcionamiento más cercano a la vida cotidiana. Postgres al ser un manejador distinto y diseñado para bases de datos muy grandes nos complicó un poco la parte de programación ya que al no estar familiarizados con su sintaxis y funcionamiento retrasó la entrega del proyecto, por eso mismo se optó por comenzar a utilizar phpMYAdmin ya que para la unión de la aplicación con la base de datos. Poder trabajar en equipo es complejo y el intercambio de ideas lleva a una retroalimentación mutua, en este caso la adaptación de dos proyectos que estaban diseñados para una conceptualización de los datos distinta llevó más tiempo de lo previsto.

Felix Flores Paul jaime

Creo que aprendimos mucho de este proyecto en cuanto aprendizaje y muchas otras. Aplicamos lo que vimos durante el semestre, incluso aplicamos un poco más con el uso de interfaces en java. Aprendí más el funcionamiento de todo, así como relacionarlo con Java Por mi caso decidí aplicar Java, ya que estoy acostumbrado desde Poo hacer proyectos así. Fue muy estresante hacer la base, ya que al ser muchas clases y relaciones entre ellas puff se vuelve complicado. Aprendimos a usar phpMyadmin por el poco tiempo que teníamos como equipo nuevo, postgres era un reto al implementarlo con Java en 4 días. Creo que un poco más de tiempo, me hubiera quedado mejor, pero estoy feliz con mi trabajo.

Jiménez García María Fernanda

El proyecto dio inicio hace algunas semanas, con un diseño similar al propuesto en el presente documento. Se realizó la actualización y la mejora de una base de datos de una papelería, con un sistema de ventas. Durante la elaboración surgieron algunos inconvenientes que se fueron solucionando progresivamente. El actual sistema está implementado en java, este mismo, se adecuó a phpmyadmin. El objetivo era simular una papelería con un sistema que nos permitiera realizar ventas y así, que permita a los clientes registrarse, seleccionar los productos, comprarlos y pagarlos. Sin duda considero las aplicaciones de lo aprendido, como elementales para cualquier tipo de sistema, dado que es necesario guardar permanentemente mucha información. En este caso particular, algunos de los beneficios serían, tener un control en el manejo de la información, de clientes, del registro de ventas, permitir emitir reportes de ventas, clientes y productos actualizados que, ayudarían a una empresa a tomar mejores decisiones a corto y a largo plazo. Considero que las condiciones y el tiempo no alcanzaron para que el equipo pudiera conectar ambos trabajos de manera eficiente. Por otra parte, SQL es un lenguaje de consulta para los sistemas de bases de datos relacionales, que no posee la potencia de los lenguajes de programación. Por ende, la familiarización con la sintaxis nos fue un poco complicada, y algunas funciones no se implementaron. Aún con esto, reitero que se aplicaron diversos conocimientos adquiridos a lo largo de la materia y por lo tanto, se reforzaron.