



**Universidad Nacional Autónoma de  
México**

**Facultad de Ingeniería**

**Bases de Datos**

**Profesor: Fernando Arreola Franco**

**Integrantes:**

**-Flores Eljure Emilio**

**-Ríos García José Eduardo**

**-Rocandio Montiel Jesus Alexis**

**Proyecto final**

**Semestre: 2021-1**

## 1. Objetivo.

EL EQUIPO TIENE COMO OBJETIVO REALIZAR LA BASE DE DATOS EN TIEMPO Y FORMA, CUMPLIENDO CON TODOS LOS REQUERIMIENTOS, PARA QUE ESTA SEA FUNCIONAL Y FÁCIL DE UTILIZAR, ASÍ CUALQUIER USUARIO PODRÁ ENTENDERLA Y HACER USO DE ELLA

## 2. Introducción.

ESTE PROYECTO ESTÁ DISEÑADO PARA UNA PAPELERÍA, LA CUAL TIENE DIFERENTES REQUERIMIENTOS, ENTRE ELLOS. INFORMACIÓN DE LOS CLIENTES, UN INVENTARIO, UN REGISTRO DE VENTAS. LA PAPELERÍA QUIERE UNA PÁGINA WEB DONDE PUEDA HACER TODO LO ANTERIOR DE FORMA FÁCIL Y VISUAL, ASÍ QUE LA PAGINA PUEDE REALIZAR VENTAS, CALCULAR EL COSTO Y PODEMOS INGRESAR ARTÍCULOS. ASÍ QUE LA PAGINA ESTA DISEÑADA PARA HACER MÁS FÁCIL LA ORGANIZACIÓN DEL STOCK Y DE VENTAS DE LA PAPELERÍA.

## 3. Solución.

EL EQUIPO HARÁ UN ANÁLISIS DEL PROBLEMA, REALIZARA DIAGRAMAS LOS CUALES LES PERMITIRÁN MODELAR LA BASE DE DATOS, CON ESTO ANALIZARA CUAL ES EL CAMINO MÁS VIABLE Y HARÁ USO DE DIFERENTES SOFTWARES. EL EQUIPO TRABAJARÁ CON LA METODOLOGÍA SCRUM, POR LO TANTO SE TENDRÁ QUE CENTRAR EN OBJETIVOS FIJADOS EN FECHAS CONCISAS, POR LO QUE LAS ACTIVIDADES SERÁN PLANEADAS Y VIGILADAS POR LOS INTEGRANTES.

## 4. Plan de trabajo.

EL EQUIPO TIENE QUE ASIGNAR FECHAS DE REUNIONES, FECHAS DE ENTREGA Y ASIGNAR ROLES, TODO ESTO PARA EL CUMPLIMIENTO DE LAS ACTIVIDADES, EN TIEMPO Y FORMA. EL MEDIO DE COMUNICACIÓN OFICIAL ES WHATSAPP Y LAS REUNIONES SE HACEN EN ZOOM, LOS DOCUMENTOS TAMBIÉN SE ALMACENAN EN WHATSAPP. PLATICAMOS UN POCO EN LA PRIMERA REUNIÓN, ESTA FUE POR WHATSAPP, AQUÍ DESCUBRIMOS LAS HABILIDADES DE CADA UNO POR LO QUE SE ASIGNARON TAREAS DE ACUERDO CON LAS HABILIDADES

ESTÁ MATRIZ ES PARA ASIGNAR RESPONSABILIDADES

MATRIZ RACI			
	EFE	JERG	JARM
MER	A,C	A,C	R
MR	A,C	R	A,C
Desarrollo bases	R	R	R
Desarrollo pagina	R	R	R
documentación	R	I	C
Pruebas	I	C	R
Análisis de problemas	I	R	C,A
Implementación del diseño	R	R	R
presentación	A,C	A,C	R

#### ACTIVIDADES A REALIZAR Y DIAS

actividad	fecha de reunion	fecha inicio actividad	duracion en dias actividad	fin actividad
roles	28/06/2021	28/06/2021	0	28/06/2021
analisis del problema	29/06/2021	29/06/2021	1	30/06/2021
analisis de recursos a utilizar	01/07/2021	01/07/2021	0	01/07/2021
busqueda de recursos y lluvia de ideas	01/07/2021	01/07/2021	0	01/07/2021
modelo entidad relacion	03/07/2021	04/07/2021	2	06/07/2021
modelo relacional	06/07/2021	07/07/2021	1	08/07/2021
analisis de la base de datos	09/07/2021	09/07/2021	5	14/07/2021
programacion de la base de datos	14/07/2021	15/07/2021	7	22/07/2021
creacion pagina web	22/07/2021	23/07/2021	5	28/07/2021
modificaciones	29/07/2021	29/07/2021	2	31/07/2021
conexión de la base de datos	31/07/2021	01/08/2021	4	04/08/2021
revision de la pagina	05/08/2021	05/08/2021	0	05/08/2021
pruebas	06/08/2021	06/08/2021	1	07/08/2021
analisis de detalles	08/08/2021	08/08/2021	2	10/08/2021
documentacion	11/08/2021	11/08/2021	2	13/08/2021
conclusiones	14/08/2021	14/08/2021	0	14/08/2021
revision final	15/08/2021	15/08/2021	0	15/08/2021

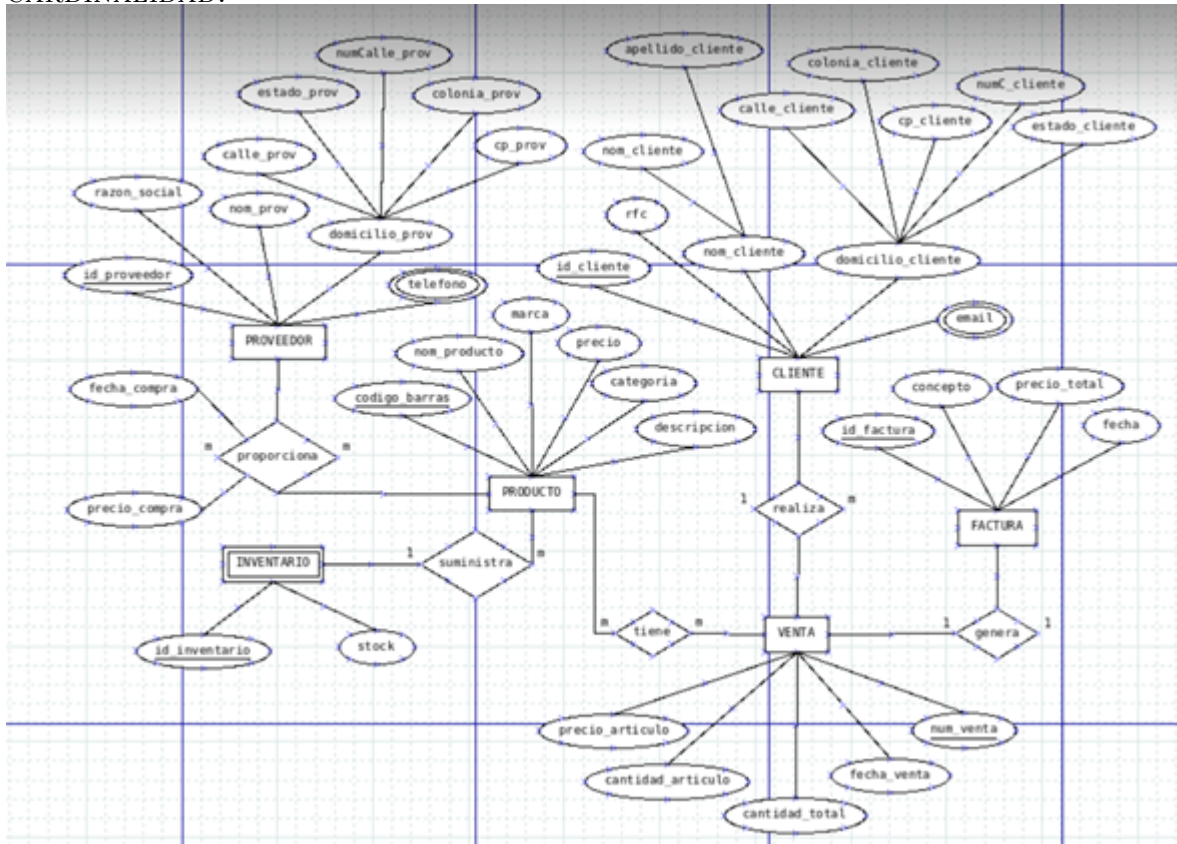
## 5. Diseño.

EN ESTE CURSO APRENDIMOS A DISEÑAR UNA BASE DE DATOS DE FORMA CORRECTA, YA QUE UN MAL DISEÑO DE ESTA PUEDE SER FATAL PARA EL PROYECTO, AQUÍ PRESENTAREMOS ALGUNOS DESEÑOS REALIZADOS POR EL EQUIPO

### 5.1. Diseño conceptual.

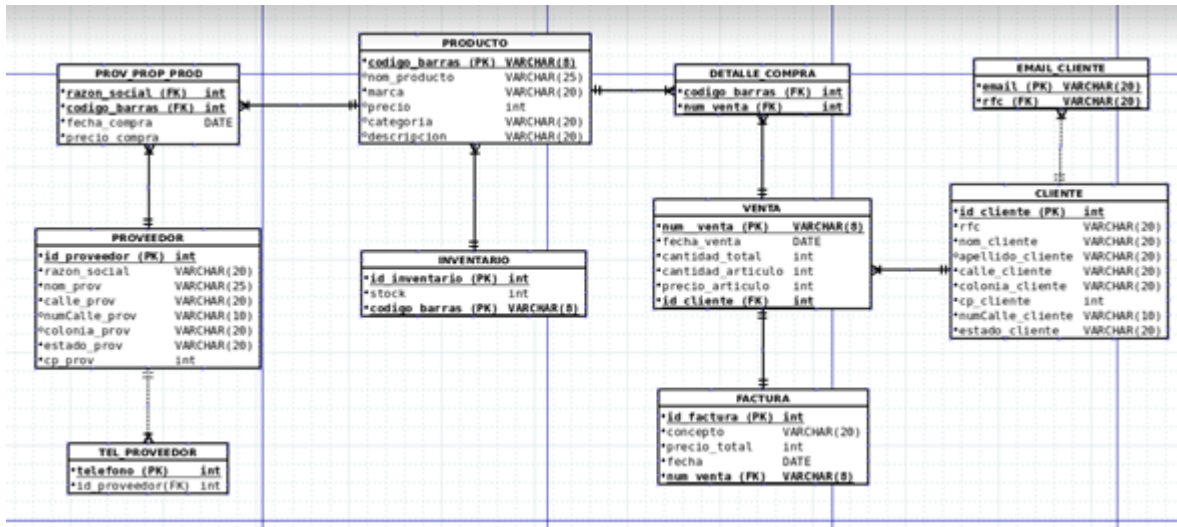
ESTE DISEÑO ES EL MÁS BÁSICO DE TODOS, YA QUE A PARTIR DE LO QUE EL CLIENTE NOS DICE NOSOTROS PARTIMOS. ASÍ PODEMOS CATEGORIZAR O ANALIZAR LO QUE EL

CLIENTE QUIERE Y PODEMOS VER CUÁL ES UNA ENTIDAD, UNA RELACIÓN, ADEMÁS DE SU CARDINALIDAD.



## 5.2. Diseño lógico.

AHORA TENEMOS QUE CREAR EL MODELO RELACIONAL A PARTIR DE NUESTRO MER



### 5.3. Diseño físico.

ES MOMENTO DE CREAR EL DISEÑO FÍSICO EN TABLAS.

```

1 -----TABLA 1-----
2 CREATE TABLE PROVEEDOR (
3 id_proveedor INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
4 razon_social VARCHAR(25),
5 nom_prov VARCHAR(25) NOT NULL,
6 calle_prov VARCHAR(20) NOT NULL,
7 colonia_prov VARCHAR(20) NOT NULL,
8 estado_prov VARCHAR(20) NOT NULL,
9 numCalle_prov VARCHAR(20) NOT NULL,
10 cp_prov INT NOT NULL,
11 CONSTRAINT pk_id_proveedor PRIMARY KEY (id_proveedor)
12 );
13
14 -----TABLA 2-----
15 CREATE TABLE TEL_PROVEEDOR (
16 telefono VARCHAR(12) NOT NULL,
17 id_proveedor INT NOT NULL,
18 CONSTRAINT pk_tel_proveedor PRIMARY KEY (telefono),
19 CONSTRAINT id_prov_tel FOREIGN KEY(id_proveedor) REFERENCES PROVEEDOR(id_proveedor)
20 );
21
22
23 -----TABLA 3-----
24 CREATE TABLE PRODUCTO (
25 codigo_barras VARCHAR(8) NOT NULL,
26 nom_producto VARCHAR(25) NOT NULL,
27 marca VARCHAR(20) NOT NULL,
28 precio DECIMAL NOT NULL,
29 categoria VARCHAR(20) NOT NULL,
30 descripcion VARCHAR(30) NOT NULL,
31 CONSTRAINT pk_codigo_barras PRIMARY KEY (codigo_barras)
32 );
33
34
35 -----TABLA 4-----
36 CREATE TABLE VENTA_PROVEEDOR (
37 id_proveedor INT NOT NULL,
38 codigo_barras VARCHAR(8) NOT NULL,
39 fecha_compra DATE NOT NULL,
40 precio_compra NUMERIC(6,2),
41 CONSTRAINT fk_id_proveedor FOREIGN KEY (id_proveedor) REFERENCES PROVEEDOR(id_proveedor),
42 CONSTRAINT fk_codigo_barras FOREIGN KEY (codigo_barras) REFERENCES PRODUCTO(codigo_barras)
43 );
44
45
46 -----TABLA 5-----
47 CREATE TABLE INVENTARIO (
48 id_inventario INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
49 codigo_barras VARCHAR(8) NOT NULL,
50 stock int NOT NULL,
51 CONSTRAINT pk_inventario PRIMARY KEY (id_inventario),
52 CONSTRAINT fk_codigo FOREIGN KEY (codigo_barras) REFERENCES PRODUCTO(codigo_barras)
53 );

```

```

56 -----TABLA 6-----
57 CREATE TABLE CLIENTE (
58 id_cliente INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
59 rfc VARCHAR(20) NOT NULL,
60 nom_cliente VARCHAR(20) NOT NULL,
61 apellido_cliente VARCHAR(20) NOT NULL,
62 calle_cliente VARCHAR(20) NOT NULL,
63 numC_cliente VARCHAR(10) NOT NULL,
64 colonia_cliente VARCHAR(20) NOT NULL,
65 cp_cliente int NOT NULL,
66 estado_cliente VARCHAR(20) NOT NULL,
67 CONSTRAINT pk_cliente PRIMARY KEY (id_cliente)
68 );
69
70
71 -----TABLA 7-----
72 CREATE TABLE EMAIL_CLIENTE (
73 email VARCHAR(20) NOT NULL,
74 id_cliente INT NOT NULL,
75 CONSTRAINT pk_email PRIMARY KEY (email),
76 CONSTRAINT fk_id_cliente FOREIGN KEY (id_cliente) REFERENCES CLIENTE(id_cliente)
77 );
78
79
80 -----TABLA 8-----
81 CREATE TABLE VENTA (
82 num_venta VARCHAR(10) NOT NULL,
83 id_cliente INT NOT NULL,
84 fecha_venta DATE NOT NULL,
85 cantidad_articulo int NOT NULL,
86 precio_articulo DECIMAL NOT NULL,
87 cantidad_total DECIMAL NOT NULL,
88 CONSTRAINT pk_num_venta PRIMARY KEY (num_venta),
89 CONSTRAINT fk_cliente_venta FOREIGN KEY (id_cliente) REFERENCES CLIENTE(id_cliente)
90 );
91
92
93 -----TABLA 9-----
94 CREATE TABLE DETALLE_COMPRA (
95 num_venta VARCHAR(10) NOT NULL,
96 codigo_barras VARCHAR(8) NOT NULL,
97 CONSTRAINT fk_codigo_barras FOREIGN KEY (codigo_barras) REFERENCES PRODUCTO(codigo_barras),
98 CONSTRAINT fk_num_venta FOREIGN KEY (num_venta) REFERENCES VENTA(num_venta)
99 );
100

```

```

102 -----TABLA 10-----
103 CREATE TABLE FACTURA (
104 id_factura INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
105 num_venta VARCHAR(10) NOT NULL,
106 fecha DATE NOT NULL,
107 concepto VARCHAR(20),
108 precio_total DECIMAL,
109 CONSTRAINT pk_factura PRIMARY KEY (id_factura),
110 CONSTRAINT fk_num_venta FOREIGN KEY (num_venta ) REFERENCES VENTA(num_venta)
111 );

```

## 6. Implementación.

```

CREATE VIEW vwProductosPorAgotarse
AS
SELECT nom_producto FROM PRODUCTO PR
INNER JOIN INVENTARIO I ON I.codigo_barras= PR.codigo_barras
WHERE I.stock <= 2;

```



```

CREATE OR REPLACE FUNCTION fTotal_Periodo(fecha_Inicio DATE, fecha_Fin DATE)
RETURNS NUMERIC LANGUAGE plpgsql
AS $$
DECLARE
    venta_Total_Periodo INT:=0;
BEGIN
    SELECT SUM(cantidad_total) FROM VENTA
    WHERE fecha_venta BETWEEN fecha_Inicio AND fecha_Fin
    INTO venta_Total_Periodo;

    RETURN venta_Total_Periodo;
END

```

```

CREATE FUNCTION fUtilidad(codigo_barras_prd VARCHAR(30))
RETURNS NUMERIC LANGUAGE plpgsql
AS $$
DECLARE
    precio_venta NUMERIC;
    ganancia NUMERIC;
    utilidad NUMERIC;
BEGIN
    SELECT precio
    FROM PRODUCTO
    WHERE codigo_barras=codigo_barras_prd
    INTO precio_venta;

    SELECT precio_compra
    FROM VENTA_PROVEEDOR
    INTO ganancia;

    utilidad:= precio_venta - ganancia;
    RETURN utilidad;
END

```

```
|CREATE VIEW vwProductosPorAgotarse  
AS  
SELECT nom_producto FROM PRODUCTO PR  
INNER JOIN INVENTARIO I ON I.codigo_barras= PR.codigo_barras  
WHERE I.stock <= 2;
```

```
CREATE TRIGGER tNewStock BEFORE INSERT ON INVENTARIO  
FOR EACH ROW  
EXECUTE PROCEDURE NewStock();
```

```

CREATE OR REPLACE FUNCTION NewStock()
RETURNS trigger LANGUAGE plpgsql
AS
$$
BEGIN
    IF stock <= 0 THEN
        ROLLBACK;
        RAISE NOTICE 'Sin productos disponibles';
    ELSE
        UPDATE inventario I SET stock = stock - cantidad
        WHERE I.precio_unitario = venta_detalle.precio_unitario;
        IF stock <= 0 THEN
            ROLLBACK;
            RAISE NOTICE 'Producto agotado';
            ELSIF unidades_stock < 2 THEN
                RAISE NOTICE 'Quedan menos de dos productos en almacen';
            END IF;
        END IF;
    END
END

```

## 6.1. Conclusiones

-ROCANDIO MONTIEL JESUS ALEXIS: EN ESTE PROYECTO SE PUSIERON A PRUEBA VARIAS HABILIDADES ADQUIRIDAS ANTERIORMENTE EN CONJUNTO CON LAS BASES DE DATOS, PERO ESTO LO COMPLICÓ UN POCO, PUES AHORA TENEMOS QUE CONECTAR LAS BASES DE DATOS A UNA INTERFAZ Y SEGUIR AGREGANDO, ELIMINANDO DATOS ENTRE OTRAS COSAS; YA NO ES COMO CUANDO COMENZÁBAMOS A PROGRAMAR QUE TODO SE ALMACENABA EN MEMORIA. EN ESTA OCASIÓN DECIDIMOS DIVIDIRNOS UN POCO EL

PROYECTO, A PESAR DE QUE SÓLO LOGRAMOS SER 3, SACAMOS ADELANTE ESTA PARTE, TANTO COMO LA PROGRAMACIÓN WEB CON ALGO DE REACT, DONDE LO ELEGIMOS YA QUE CREÍMOS QUE SERÍA MÁS FÁCIL LA CONEXIÓN A LAS BASES DE DATOS, POR QUE CON ANTERIORIDAD LOGRAMOS TENER UNA INTERFAZ PERO NO LOGRAMOS CONECTAR LAS BASES, ESTO FUE EN HTML Y PHP, ALGO RARO PASABA, ENTONCES DECIDIMOS CAMBIAR A REACT, NO FUE MUCHO EL CAMBIO, PERO SI EN PROCESOS, YA QUE USAMOS LÓGICA BACK END. TENEMOS ACCESO A TODOS LOS DATOS, PERO OBSERVAMOS MÁS CONVENIENTE CONECTAR CÓDIGO QUE CORRER SCRIPS. FUE ALGO LABORIOSO SINCERAMENTE PARA TRES PERSONAS, PERO SE HIZO LOGRO EL OBJETIVO DE APRENDER A USAR MÁS LAS BASES DE DATOS, Y DE REALIZAR UNA CONEXIÓN, LA CUAL NO FUE TAMPOCO SENCILLO, PERO SI QUE CORRIERON LOS DATOS A DIFERENCIA DE NUESTRA PÁGINA ANTERIOR.

-RÍOS GARCÍA JOSÉ EDUARDO: LA ELABORACIÓN DEL PRESENTE PROYECTO FUE UN PROCESO DE LARGO APRENDIZAJE YA QUE PUDIMOS PROYECTAR EL CONOCIMIENTO ADQUIRIDO A LO LARGO DEL CURSO DE LA ASIGNATURA BASE DE DATOS. TUVIMOS COMPLICACIONES DEL LADO DE LA INTERFAZ YA QUE OPTAMOS POR UNA TECNOLOGÍA IMPLEMENTADA EN PHP PERO NOS SURGIERON VARIOS PROBLEMAS LOS CUALES PUDIMOS RESOLVER CON EL FRAMEWORK REACT YA QUE CON ESTE PUDIMOS IMPLEMENTAR EL USO DE USUARIOS Y PODER TENER UNA INTERFAZ AMIGABLE PARA EL USUARIO FINAL. SE CUMPLIERON CON CADA UNO DE LOS PUNTOS ESTABLECIDOS EN EL DOCUMENTO DONDE ESTABAN DESCRITOS LOS REQUERIMIENTOS. LA REALIZACIÓN DE ESTE PROYECTO FUE UN TANTO SATISFACTORIO YA QUE SE PIENSA A FUTURO PODER COMERCIALIZAR DICHO PRODUCTO, BASÁNDONOS EN LAS NECESIDADES DE LOS CLIENTES QUE PODAMOS ADQUIRIR.

-FLORES ELJURE EMILIO: ESTE PROYECTO ES UNO DE LOS MAS INTERESANTES Y DIFICILES QUE HE HECHO EN MI VIDA, YA QUE MANEJE MUCHOS CONCEPTOS, SOBRE TODO DE BASES DE DATOS, DE SERVIDORES, ETC. EN EL CAMINO HUBIERON MUCHAS DIFICULTADES, PERO CON MI EQUIPO LOGRE RESOLVER MUCHAS DE ELLAS, AUQNUE TARDMAOS MUCHO EN REALIZAR ESTE PROYECTO, AL PRINCIPIO NO SABIAMOS MUCHO, PERO CONFORME AVANZABAMOS IBAMOS COMPRENDIENDO TEMAS