



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Asignatura: **Bases de Datos**

Profesor: **Ing. Fernando Arreola Franco**

Proyecto Final

Equipo: **RAJAVX**

Alumnos:

Jimenez Avila Javier Alejandro
Mora Castro Fernando Alexis
Resendiz Cruz Rodrigo Daniel
Valdelamar Tamez Valeria

Grupo: **1°**

Semestre: **2022-1**

Fecha: **11 de Diciembre 2021**

Introducción

Planteamiento

Este proyecto tiene el fin de realizar una base de datos para la administración de una cadena de papelerías, que tenga almacenado los datos de los proveedores tales como; la razón social, domicilio, nombre y teléfonos de los proveedores, rfc, nombre, domicilio y al menos un email de los clientes, el inventario de productos en venta 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, existencias), los datos que se desean guardar de los diferentes artículos son; 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, de las ventas; 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 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 el pedido se completa pero quedan menos de 3 en stock, se deberá emitir una alerta.
- Dada una fecha, o una fecha de inicio y fecha de fin, regresar la cantidad total que se vendió y la ganancia correspondiente en esa fecha/periodo.
- Permitir obtener el nombre de aquellos productos de los cuales hay menos de 3 en stock.
- Generar de forma automática la factura de una compra.
- Realizar los índices que se requieran.

Deberán tomarse las siguientes consideraciones:

- El número de venta debe tener un formato similar a "VENT-001".
- Donde esté presente el atributo domicilio, este debe estar compuesto por estado, código postal, colonia, calle y número.

Objetivos

1. Aplicar los conceptos y técnicas vistas en clase para diseñar y crear una base de datos y su administración para una cadena de papelerías, que cumpla con las características propias del solicitante.
2. Hacer buen uso del tiempo para poder desarrollar y solucionar nuestra problemática.

Propuesta de Resolución

Fue enumerada en una serie de 11 pasos a seguir:

1. Empezar con el diseño de nuestra base de datos, en primera instancia la lectura y comprensión de lo que se nos pide.
2. Organizar la participación del equipo de trabajo así como la asignación de tareas que le tocará a cada integrante.
3. La puntualización de reuniones necesarias para ir revisando el avance del proyecto.
4. Realizar el diagrama entidad relación de nuestra base de datos,y después el diagrama relacional.
5. La creación de la base de datos y sus tablas (campos, relaciones y restricciones) en el manejador.
6. Programación SQL sobre los requerimientos dados.
7. Diseño y Programación de la interfaz gráfica.
8. Conexión de nuestra base de datos con la interfaz gráfica, y pruebas de su funcionamiento.
9. Elaboración de la documentación y manual de usuario.
10. Pruebas finales de nuestra base de datos y de su adecuado funcionamiento.
11. Preparación de la exposición del proyecto.

Plan de Trabajo

Se decidió organizar todas nuestras actividades por medio de un diagrama de Gantt:

Nº Actividad	Actividad	Inicio	Final	01-nov	03-nov	05-nov	07-nov	09-nov	11-nov	13-nov	15-nov	17-nov	19-nov	21-nov	23-nov	25-nov	27-nov	29-nov	01-dic	03-dic	05-dic	07-dic	09-dic	11-dic	13-dic
1	Planteamiento de Objetivos del Proyecto	01/11/2021	03/11/2021																						
2	Análisis de requerimientos	02/11/2021	05/11/2021																						
3	Asignación de actividades	05/11/2021	07/11/2021																						
4	Elaboración del MER	07/11/2021	13/11/2021																						
5	Elaboración del MR	13/11/2021	15/11/2021																						
6	Creación Base de Datos	17/11/2021	21/11/2021																						
7	Restricciones BD	19/11/2021	22/11/2021																						
8	Programación	22/11/2021	24/11/2021																						
9	Investigación Entorno Gráfico	19/11/2021	25/11/2021																						
10	Desarrollo interfaz gráfica	25/11/2021	09/12/2021																						
11	Conexión BD con Interfaz	05/12/2021	09/12/2021																						
12	Documentación en LATEX	01/12/2021	11/12/2021																						
13	Desarrollo de la presentación	07/12/2021	12/12/2021																						
14	Ensayo presentación	13/12/2021	13/12/2021																						
15	Entrega del Proyecto	28/11/2021	30/11/2021																						
16	Presentación del Proyecto	29/11/2021	01/12/2021																						

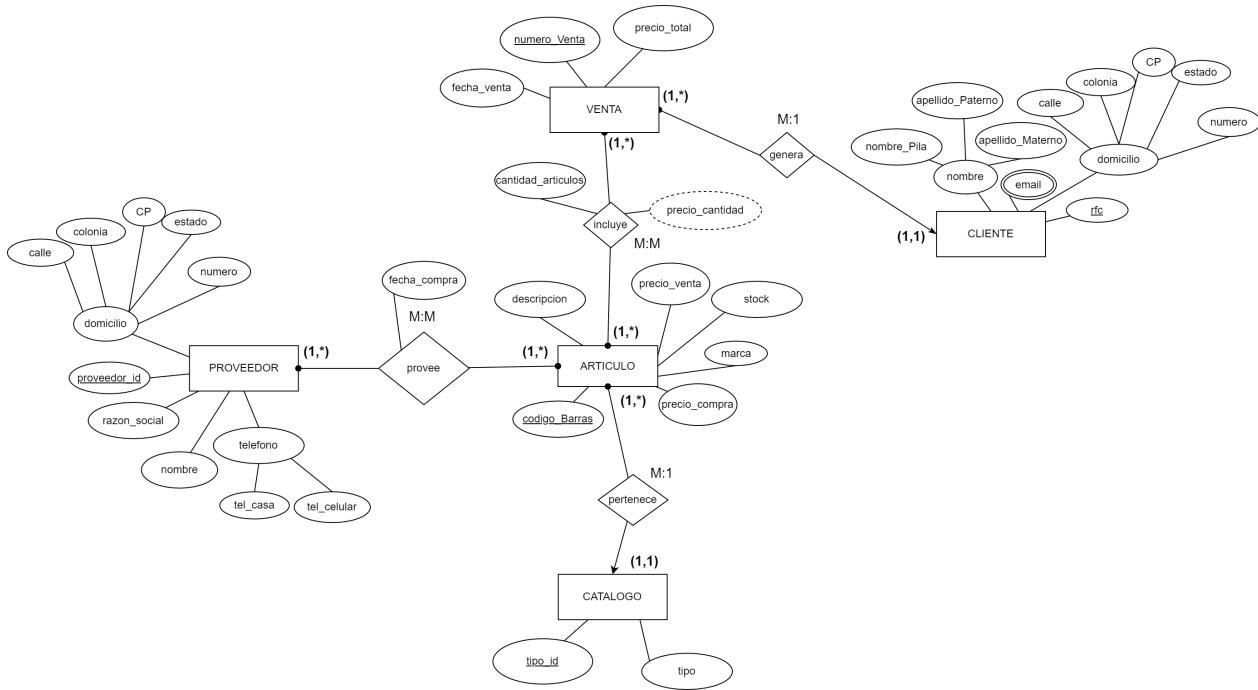
La organización de actividades por parte del equipo fue la siguiente:

Actividad	Responsable
Planteamiento de Objetivos del Proyecto	Todos los integrantes
Análisis de requerimientos	Todos los integrantes
Asignación de actividades	Resendiz Cruz Rodrigo Daniel
Elaboración del MER	Jimenez Avila Javier Alejandro
Elaboración del MR	Jimenez Avila Javier Alejandro
Creación Base de Datos	Resendiz Cruz Rodrigo Daniel
Restricciones BD	Resendiz Cruz Rodrigo Daniel
Programación	Valdelamar Tamez Valeria
Investigación Entorno Gráfico	Valdelamar Tamez Valeria
Desarrollo interfaz gráfica	Mora Castro Fernando Alexis
Conexión BD con Interfaz	Resendiz Cruz Rodrigo Daniel
Documentación en LATEX	Jimenez Avila Javier Alejandro
Desarrollo de la presentación	Jimenez Avila Javier Alejandro
Ensayo presentación	Todos los integrantes
Entrega del Proyecto	Resendiz Cruz Rodrigo Daniel
Presentación del Proyecto	Todos los integrantes

Diseño

Realizamos el primer bosquejo de la base de datos, posteriormente se realizó la afinación del diseño y estructura de esta misma por medio del diagrama entidad relación.

Modelo Entidad Relación:



Descripción:

Nuestro Modelo Entidad Relación fue hecho con base a los requerimientos dados. Empezaremos explicando la tabla Proveedor que es una subtabla de la tabla artículo, esta tabla guardará los datos del proveedor, consta de 5 atributos principales; domicilio (que se descompone en un subtabla con los atributos; calle, colonia, cp, estado y número), proveedor_id, razón_social, nombre y teléfono (siendo una subtabla con los campos tel_casa y tel_celular), en el caso del atributo teléfono elegimos que fuera también una tabla, para evitar que fuera multivalorado, lo cual se pensó de forma estratégica, ya que no lo vimos necesario. Su cardinalidad según lo analizado es de 1:M en relación con la tabla del artículo.

Después tenemos la tabla artículo que está compuesta de 6 atributos, los cuales son;

descripción, precio_venta, stock (existencias), marca, precio_compra y código_barras. El precio de venta es a lo que la papelería venderá el artículo y el precio_compra es a lo que lo compró la papelería. Esta tabla está unida con tres tablas (subtablas); una llamada venta con cardinalidad 1:M, después con dos tablas subyacentes; la de proveedor que ya la mencionamos con anterioridad, con cardinalidad 1:M, y la tabla catálogo, con cardinalidad 1:1 que explicaremos a continuación:

La tabla catálogo alberga los tipos de productos que existirán en el almacén, cuenta con solo dos atributos; tipo_id y tipo, el número y nombre del artículo.

La tabla venta, subtabla de artículo y de la tabla cliente, esta tabla tiene el fin de guardar el número de artículos comprados, su precio unitario, el precio total y la fecha de la compra, tiene 5 atributos puntuizando, los cuales son; fecha_venta, numero_venta, precio_total, cantidad_articulos y precio_cantidad. En la relación venta-articulos tiene una cardinalidad de 1:M y la relación con la tabla cliente su cardinalidad es de 1:M – M:1 y de 1:1.

Normalización

- Primera Forma normal: sí cumple, ya que no presenta grupos de repetición y cada columna contienen valores atómicos.
- Segunda Forma Normal: De terminaremos si existen relaciones parciales con base al diagrama relacional:

En la tabla CLIENTE: X → {A,B,C,D,E,F}

En la tabla CLIENTE_EMAIL: A → {Y}

En la tabla VENTA: G → {H,I,J,A}

En la tabla VENTA_ALMACEN: {G,L} → {M,N,O}

En la tabla ARTICULO: L → {K,R,S,T,W}

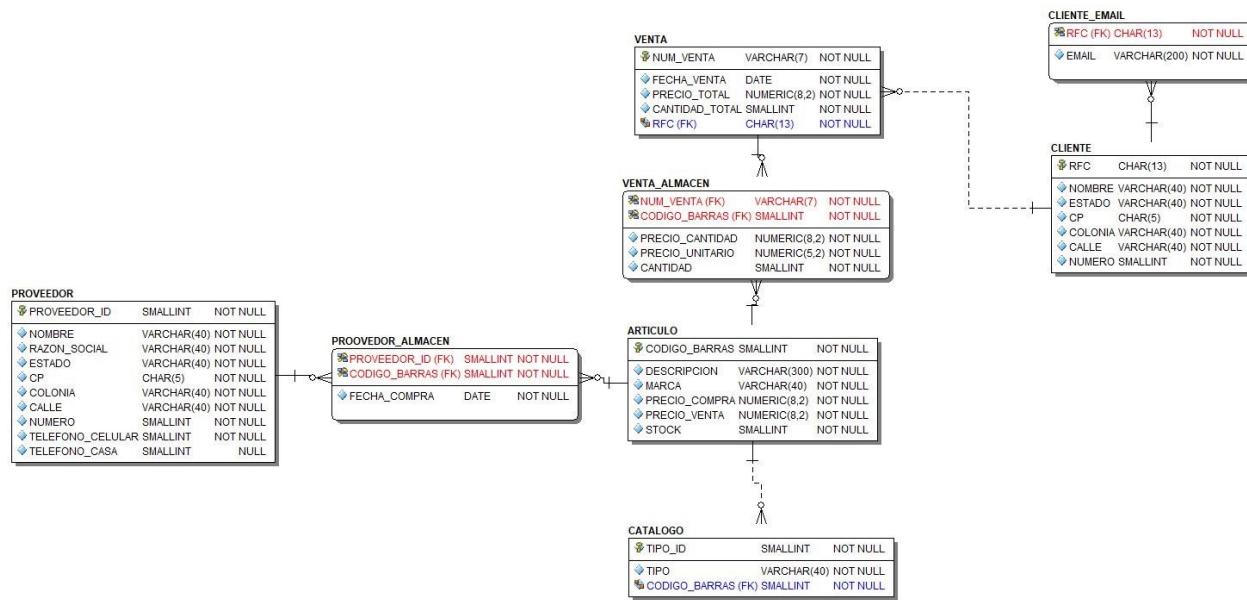
En la tabla PROVEDOR_ALMACEN: {Z,L} → {AA}

En la tabla PROVEDOR: Z → {BB,CC,DD,EE,FF,GG,HH,II,JJ}

En la tabla CATALOGO: L → {PP}

- Tercera Forma Normal: Sí cumple ya que se encuentra en 2FN y cualquier atributo no -principal de la tabla no es transitivamente dependiente de cada clave candidata de la misma.

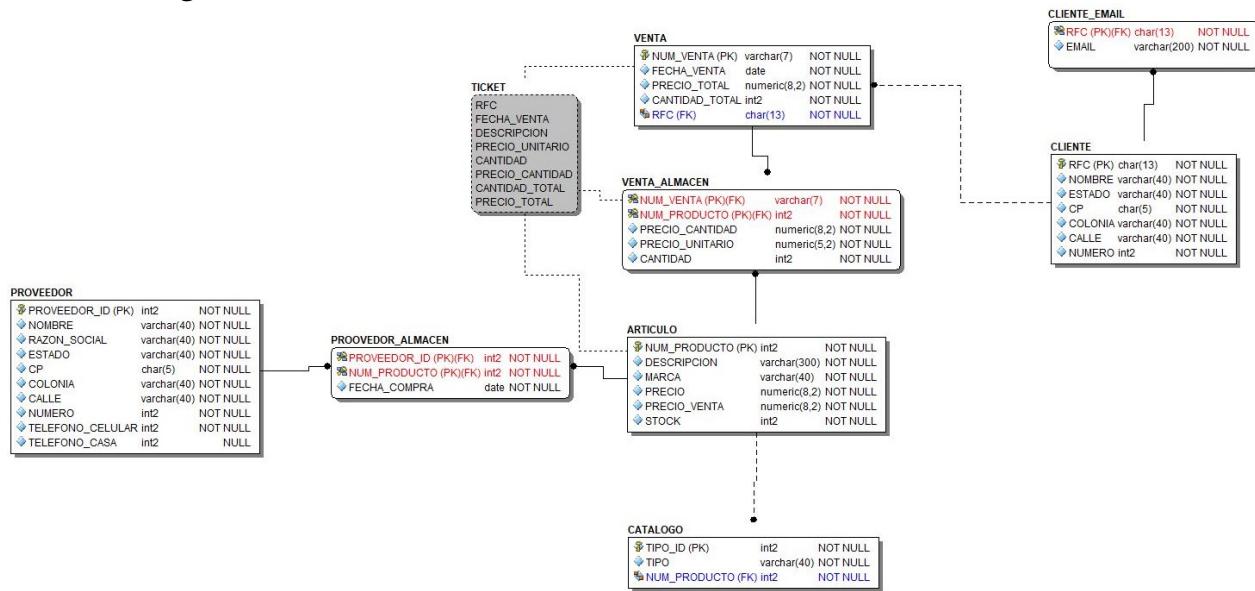
Modelo Relacional:



Descripción:

Para el modelo relacional se agregaron las tablas “proveedor_almacen” y la tabla “venta_almacen” las cuales van a guardar un registro detallado de las ventas que se efectúen y de las compras de los productos. Adicionalmente se creó una tabla llamada “cliente_email” el cual se encargará de almacenar los email de los clientes, en esa tabla.

Modelo Lógico:



Descripción:

En el modelo lógico se tiene lo mismo que en el modelo relacional, pero, se agrega la vista llamada como ticket que se nos solicita generar mediante el registro de las ventas.

Implementación

Funcionamiento los bloques de programación:

```

6 ----- ===== Consulta 1 =====
7 ----- =====
8 ----- =====
9  SELECT PRECIO_VENTA - PRECIO_COMPRA FROM ARTICULO

```

Explicación: Se toma de cada artículo el precio de compra y el precio de venta, estos dos se restan y la consulta regresa el resultado de esta operación matemática, siendo esta la utilidad de cada artículo.

```

11 ----- =====
12 ----- ===== Consulta 2 =====
13 -----
14 CREATE TRIGGER DECREMENTO_STOCK FOR ARTICULO
15     ACTIVE BEFORE INSERT
16     POSITION 0
17 AS
18 BEGIN
19     if (Select stock from ARTICULO where NUM_PRODUCTO=@NUM_PRODUCTO) >= @Cantidad begin
20         UPDATE ARTICULO
21         SET stock = stock - @cantidad
22         WHERE NUM_PRODUCTO = @NUM_PRODUCTO
23     end
24     else
25         Select 'La cantidad supera el stock actual' AS Mensaje
26 END

```

Explicación: Se crea un trigger que se dispara antes del insert cuando esté se dispara se buscará el stock del artículo en cuestión mediante su número de producto, si este stock es mayor a la cantidad 0 se actualizará el stock de el producto encontrado, el stock actualizado será stock menos cantidad, de lo contrario si el stock es menor a 0 se regresa una alerta.

```

29 ----- =====
30 ----- ===== Consulta 3 =====
31 -----
32 select (aa.precio_venta*sum(cantidad))-(aa.precio_compra*sum(cantidad)) as ganancia
33 from venta_almacen ar
34 inner join articulo aa
35 on aa.num_producto=ar.num_producto
36 inner join venta v
37 on v.num_venta=ar.num_venta
38 where fecha_venta between '25-10-2021' and '30-10-2021'
39 group by aa.num_producto;

```

Explicación: De la venta del almacén se obtiene su ganancia, precio venta por cantidad menos precio compra por cantidad, teniendo la ganancia se busca el artículo de esta respectiva ganancia mediante un join usando el número de producto, ahora bien se realiza un segundo join para encontrar la fecha de la venta, este segundo join se realiza mediante el número de venta; finalmente los resultados se agrupan mediante el número de producto.

```

41 ----- =====
42 ----- ===== Consulta 4 =====
43 ----- =====
44
45 SELECT DESCRIPCION FROM ARTICULO WHERE STOCK > 3
46 ----- =====

```

Explicación: Se busca la descripción de los artículos donde su stock es mayor a 3.

```

46 ----- =====
47 ----- ===== Consulta 5 =====
48 ----- =====
49 CREATE VIEW TICKET AS
50 SELECT VE.RFC, VE.FECHA_VENTA, AR.DESCRIPCION, VEN.PRECIO_UNITARIO, VEN.CANTIDAD,
51 VEN.PRECIO_CANTIDAD, VE.CANTIDAD_TOTAL, VE.PRECIO_TOTAL
52 FROM VENTA VE, VENTA_ALMACEN VEN, ARTICULO AR
53 WHERE VEN.NUM_VENTA = VE.NUM_VENTA
54 AND VEN.NUM_PRODUCTO = AR.NUM_PRODUCTO

```

Explicación: Se crea una vista llamada ticket que busca rfc, fecha de venta, descripción, precio unitario, cantidad, precio cantidad, cantidad total y precio total de las tablas venta, venta almacén y artículo usando el número de venta y número de producto como filtros; esta información obtenida es usada para la generación de facturas.

La generación de la vista nos permite realizar consultas más complejas sobre esta información ya que genera una tabla virtual, por el contrario de estar consultando toda la información mediante selects.

DDL:

El ddl contiene la definición de las tablas para ser implementadas dentro de la base de datos, la creación de las tablas se muestra a continuación:

Tabla producto:

```

7   CREATE TABLE ARTICULO(
8
9     NUM_PRODUCTO    int2          NOT NULL,
10
11    DESCRIPCION     varchar(300)   NOT NULL,
12
13    MARCA           varchar(40)    NOT NULL,
14
15    PRECIO_COMPRA    numeric(8, 2)  NOT NULL,
16
17    PRECIO_VENTA    numeric(8, 2)  NOT NULL,
18
19    STOCK           int2          NOT NULL,
20
21    CONSTRAINT PK4 PRIMARY KEY (NUM_PRODUCTO),
22    CONSTRAINT CHECK_STOCK CHECK(STOCK>=0),
23    CONSTRAINT CHECK_PRECIO CHECK (PRECIO_COMPRA>0 or PRECIO_VENTA >0)
24  );

```

En la tabla producto se tiene la definición de las columnas que se van a emplear y se agregan como constraints la llave primaria la cual es el número de producto, un check que verifica que el precio de compra sea mayor que 0 y el stock sea mayor o igual a 0, se permite que el stock sea igual a cero porque pueden acabarse los artículos en la papelería, pero siguen estando en el catálogo de productos.

Tabla Catalogo:

```

34  CREATE TABLE CATALOGO(
35
36    TIPO_ID         int2          NOT NULL,
37
38    TIPO            varchar(40)   NOT NULL,
39
40    NUM_PRODUCTO    int2          NOT NULL,
41
42    CONSTRAINT PK13 PRIMARY KEY (TIPO_ID)
43
44  );

```

En la tabla catálogo se añadió el identificador para cada tipo de artículo con respecto a cada producto registrado.

Tabla Cliente:

```

54   CREATE TABLE CLIENTE(
55
56     RFC      char(13)      NOT NULL,
57
58     NOMBRE   varchar(40)   NOT NULL,
59
60     APELLIDO_PATERNO varchar(40)   NOT NULL,
61
62     APELLIDO_MATERNO varchar(40),
63
64     ESTADO   varchar(40)   NOT NULL,
65
66     CP       char(5)       NOT NULL,
67
68     COLONIA  varchar(40)   NOT NULL,
69       CALLE   varchar(40)   NOT NULL,
70
71     NUMERO   int2          NOT NULL,
72
73
74     CONSTRAINT PK1 PRIMARY KEY (RFC)
75
76   );

```

En la tabla cliente se definen los campos del nombre completo desglosado y los datos del domicilio del cliente. El RFC del cliente tiene que ser de 13 caracteres.

Tabla Cliente_Email:

```

86   CREATE TABLE CLIENTE_EMAIL(
87
88     RFC      char(13)      NOT NULL,
89
90     EMAIL   varchar(200)   NOT NULL,
91
92     CONSTRAINT PK14 PRIMARY KEY (RFC)
93
94   );

```

La tabla cliente_email es la que se encarga de almacenar los correos de cada cliente, este toma como parámetros el RFC del cliente y su email.

Tabla proveedor_almacen:

```

104   CREATE TABLE PROVEEDOR_ALMACEN(
105     PROVEEDOR_ID  int2      NOT NULL,
106     NUM_PRODUCTO int2      NOT NULL,
107     FECHA_COMPRA date     NOT NULL,
108     CONSTRAINT PK12 PRIMARY KEY (PROVEEDOR_ID, NUM_PRODUCTO)
109   );
110
111
112
113
114

```

En la tabla proveedor almacén se tienen los datos asociados a la compra de los artículos con una llave primaria compuesta de el id del proveedor al que se le compró junto con el artículo comprado, como columna que no es llave primaria se tiene la fecha de compra de dicho producto a tal proveedor.

Tabla Proveedor:

```

124   CREATE TABLE PROVEEDOR(
125     PROVEEDOR_ID      int2      NOT NULL,
126     NOMBRE            varchar(40) NOT NULL,
127     RAZON_SOCIAL      varchar(40) NOT NULL,
128     ESTADO             varchar(40) NOT NULL,
129     CP                 char(5)    NOT NULL,
130     COLONIA            varchar(40) NOT NULL,
131     CALLE              varchar(40) NOT NULL,
132
133
134
135
136
137
138
139     NUMERO            int2      NOT NULL,
140     TELEFONO_CELULAR  varchar(15) NOT NULL,
141     TELEFONO_CASA     varchar(15),
142     CONSTRAINT PK2 PRIMARY KEY (PROVEEDOR_ID)
143
144
145
146
147
148

```

La tabla de proveedor es similar al del cliente, solo que se añaden los campos de razón social y los teléfonos asociados a dicho proveedor.

Tabla Venta

```

158   CREATE TABLE VENTA(
159
160     NUM_VENTA      varchar(7)      NOT NULL,
161
162     FECHA_VENTA    date          NOT NULL,
163
164     PRECIO_TOTAL    numeric(8, 2)  NOT NULL,
165
166     CANTIDAD_TOTAL int2          NOT NULL,
167
168     RFC            char(13)       NOT NULL,
169
170     CONSTRAINT PK10 PRIMARY KEY (NUM_VENTA),
171     CONSTRAINT CHECK_NUM_VENTA CHECK (substring(NUM_VENTA,1,5)='VENT-')
172 );

```

En la tabla venta se tienen los atributos de el número de la venta, la fecha de la venta, el precio total, la cantidad total y el rfc del cliente que hace la compra. Nótese que se agregó un constraint de tipo check que revisa si el formato de la venta es similar a “VENT-001”.

Tabla venta almacen:

```

183   CREATE TABLE VENTA_ALMACEN(
184
185     NUM_VENTA      varchar(7)      NOT NULL,
186
187     NUM_PRODUCTO   int2          NOT NULL,
188
189     PRECIO_UNITARIO numeric(8, 2) NOT NULL,
190
191     CANTIDAD       int2          NOT NULL,
192
193     PRECIO_CANTIDAD numeric(8, 2) NOT NULL,
194
195     CONSTRAINT PK11 PRIMARY KEY (NUM_VENTA, NUM_PRODUCTO)
196
197 );

```

En la tabla venta almacén se agregaron los campos necesarios para registrar de manera más detallada el contenido de los artículos involucrados en una venta. La llave primaria de esta tabla es compuesta ya que se necesitan de los campos del número de venta y el número de producto para conocer el precio del producto involucrado, su cantidad y su precio de acuerdo a esa cantidad.

Llaves foráneas:

```

--- 214 ALTER TABLE CATALOGO ADD CONSTRAINT "RefARTICULO241"
215
216     FOREIGN KEY (NUM_PRODUCTO)
217
218     REFERENCES ARTICULO(NUM_PRODUCTO);
219
220     --
221
222     -- TABLE: "CLIENTE_EMAIL"
223
224     --
225
226
227
228
229 ALTER TABLE CLIENTE_EMAIL ADD CONSTRAINT RefCLIENTE251
230
231     FOREIGN KEY (RFC)
232
233     REFERENCES CLIENTE(RFC);
234
235     --
236
237     -- TABLE: "PROVEEDOR_ALMACEN"
238
239     --
240
241
242
243
244 ALTER TABLE PROVEEDOR_ALMACEN ADD CONSTRAINT RefPROVEEDOR211
245
246     FOREIGN KEY (PROVEEDOR_ID)
247
248     REFERENCES PROVEEDOR(PROVEEDOR_ID);
249
250
251
252 ALTER TABLE PROVEEDOR_ALMACEN ADD CONSTRAINT RefARTICULO221
253
254     FOREIGN KEY (NUM_PRODUCTO)
255
256     REFERENCES ARTICULO(NUM_PRODUCTO);
257
258     --
259
260     -- TABLE: "VENTA"
261
262     --
263
264
265
266
267 ALTER TABLE VENTA ADD CONSTRAINT RefCLIENTE261
268
269     FOREIGN KEY (RFC)
270
271     REFERENCES CLIENTE(RFC) ON UPDATE CASCADE;

```

```

277  --
278
279 ALTER TABLE VENTA_ALMACEN ADD CONSTRAINT RefVENTA141
280
281     FOREIGN KEY (NUM_VENTA)
282
283     REFERENCES VENTA(NUM_VENTA) ON UPDATE CASCADE;
284
285
286
287 ALTER TABLE VENTA_ALMACEN ADD CONSTRAINT RefARTICULO151
288
289     FOREIGN KEY (NUM_PRODUCTO)
290
291     REFERENCES ARTICULO(NUM_PRODUCTO);

```

Una vez hechas las tablas se crearon los constraints de llaves foráneas que las enlazan.

Secuencia venta:

```

299   Create sequence seq_venta
300     increment by 1
301     minvalue 1
302     no maxvalue
303     start with 1
304     cache 3
305     no cycle;

```

La secuencia venta es una secuencia que se utiliza para obtener el número que va después de “VENT-” la secuencia inicia en 1, su valor mínimo es 1, no tienen valor máximo, inicia en 1 con tres valores en caché y es una secuencia no cíclica.

Indice nombres empleados:

```

308   create index idx_nombres_clientes
309   on cliente(NOMBRE,APELLIDO_PATERNO,APELLIDO_MATERO);
310
311

```

El modelo DDL completo se muestra a continuación:

```

1  --
2
3  -- TABLE: "ARTICULO"
4
5  --
6
7  CREATE TABLE ARTICULO(
8
9      NUM_PRODUCTO    int2          NOT NULL,
10
11     DESCRIPCION     varchar(300)   NOT NULL,
12
13     MARCA          varchar(40)    NOT NULL,
14
15     PRECIO_COMPRA    numeric(8, 2)  NOT NULL,
16
17     PRECIO_VENTA    numeric(8, 2)  NOT NULL,
18
19     STOCK           int2          NOT NULL,
20
21     CONSTRAINT PK4 PRIMARY KEY (NUM_PRODUCTO),
22     CONSTRAINT CHECK_STOCK CHECK(STOCK>=0),
23     CONSTRAINT CHECK_PRECIO CHECK (PRECIO_COMPRA>0 or PRECIO_VENTA >0)
24 );
25
26
27
28 -- TABLE: "CATALOGO"
29
30
31
32
33
34 CREATE TABLE CATALOGO(
35
36     TIPO_ID         int2          NOT NULL,
37
38     TIPO            varchar(40)   NOT NULL,
39
40     NUM_PRODUCTO    int2          NOT NULL,
41
42     CONSTRAINT PK13 PRIMARY KEY (TIPO_ID)
43 );
44
45
46
47
48 -- TABLE: "CLIENTE"
49
50
51
52
53
54 CREATE TABLE CLIENTE(
55
56     RFC              char(13)     NOT NULL,
57
58     NOMBRE           varchar(40)   NOT NULL,
59
60     APELLIDO_PATERNO varchar(40)   NOT NULL,
61
62     APELLIDO_MATERNO varchar(40),
63
64     ESTADO           varchar(40)   NOT NULL,
65
66     CP               char(5)      NOT NULL,
67
68     COLONIA          varchar(40)   NOT NULL,
69

```

```

70      CALLE      varchar(40)      NOT NULL,
71
72      NUMERO     int2          NOT NULL,
73
74      CONSTRAINT PK1 PRIMARY KEY (RFC)
75
76  );
77
78  --
79
80  -- TABLE: "CLIENTE_EMAIL"
81
82  --
83
84
85
86  CREATE TABLE CLIENTE_EMAIL(
87
88      RFC      char(13)      NOT NULL,
89
90      EMAIL    varchar(200)    NOT NULL,
91
92      CONSTRAINT PK14 PRIMARY KEY (RFC)
93
94  );
95
96  --
97
98  -- TABLE: "PROVEEDOR_ALMACEN"
99
100 --
101
102
103
104  CREATE TABLE PROVEEDOR_ALMACEN(
105
106      PROVEEDOR_ID  int2      NOT NULL,
107
108      NUM_PRODUCTO int2      NOT NULL,
109
110      FECHA_COMPRA date      NOT NULL,
111
112      CONSTRAINT PK12 PRIMARY KEY (PROVEEDOR_ID, NUM_PRODUCTO)
113
114  );
115
116  --
117
118  -- TABLE: "PROVEEDOR"
119
120  --
121
122
123
124  CREATE TABLE PROVEEDOR(
125
126      PROVEEDOR_ID  int2      NOT NULL,
127
128      NOMBRE       varchar(40)  NOT NULL,
129
130      RAZON_SOCIAL varchar(40)  NOT NULL,
131
132      ESTADO        varchar(40)  NOT NULL,
133
134      CP            char(5)     NOT NULL,
135
136      COLONIA      varchar(40)  NOT NULL,
137
138      CALLE        varchar(40)  NOT NULL,

```

```

139
140     NUMERO          int2      NOT NULL,
141
142     TELEFONO_CELULAR  varchar(15)  NOT NULL,
143
144     TELEFONO_CASA    varchar(15),
145
146     CONSTRAINT PK2 PRIMARY KEY (PROVEEDOR_ID)
147
148 );
149
150 --
151
152 -- TABLE: "VENTA"
153
154 --
155
156
157
158 CREATE TABLE VENTA(
159
160     NUM_VENTA      varchar(7)      NOT NULL,
161
162     FECHA_VENTA   date      NOT NULL,
163
164     PRECIO_TOTAL   numeric(8, 2)  NOT NULL,
165
166     CANTIDAD_TOTAL int2      NOT NULL,
167
168     RFC            char(13)      NOT NULL,
169
170     CONSTRAINT PK10 PRIMARY KEY (NUM_VENTA),
171     CONSTRAINT CHECK_NUM_VENTA CHECK (substring(NUM_VENTA,1,5)='VENT-')
172 );
173
174
175 --
176
177 -- TABLE: "VENTA_ALMACEN"
178
179 --
180
181
182
183 CREATE TABLE VENTA_ALMACEN(
184
185     NUM_VENTA      varchar(7)      NOT NULL,
186
187     NUM_PRODUCTO   int2      NOT NULL,
188
189     PRECIO_UNITARIO numeric(8, 2)  NOT NULL,
190
191     CANTIDAD       int2      NOT NULL,
192
193     PRECIO_CANTIDAD numeric(8, 2)  NOT NULL,
194
195
196     CONSTRAINT PK11 PRIMARY KEY (NUM_VENTA, NUM_PRODUCTO)
197
198 );
199
200
201 --
202
203 -- VIEW: "TICKET"
204
205 --
206
207

```

```
208 -- TABLE: "CATALOGO"
209
210 --
211
212
213
214 ALTER TABLE CATALOGO ADD CONSTRAINT "RefARTICULO241"
215
216     FOREIGN KEY (NUM_PRODUCTO)
217
218     REFERENCES ARTICULO (NUM_PRODUCTO);
219
220 --
221 --
222
223 -- TABLE: "CLIENTE_EMAIL"
224
225 --
226
227
228
229 ALTER TABLE CLIENTE_EMAIL ADD CONSTRAINT RefCLIENTE251
230
231     FOREIGN KEY (RFC)
232
233     REFERENCES CLIENTE (RFC);
234
235 --
236 --
237
238 -- TABLE: "PROVEEDOR_ALMACEN"
239
240 --
241
242
243
244 ALTER TABLE PROVEEDOR_ALMACEN ADD CONSTRAINT RefPROVEEDOR211
245
246     FOREIGN KEY (PROVEEDOR_ID)
247
248     REFERENCES PROVEEDOR (PROVEEDOR_ID);
249
250
251
252 ALTER TABLE PROVEEDOR_ALMACEN ADD CONSTRAINT RefARTICULO221
253
254     FOREIGN KEY (NUM_PRODUCTO)
255
256     REFERENCES ARTICULO (NUM_PRODUCTO);
257
258 --
259 --
260
261 -- TABLE: "VENTA"
262
263 --
264
265
266
267 ALTER TABLE VENTA ADD CONSTRAINT RefCLIENTE261
268
269     FOREIGN KEY (RFC)
270
271     REFERENCES CLIENTE (RFC) ON UPDATE CASCADE;
272
273 --
274
275 -- TABLE: "VENTA_ALMACEN"
276
```

```
277    --
278
279 ALTER TABLE VENTA_ALMACEN ADD CONSTRAINT RefVENTA141
280
281     FOREIGN KEY (NUM_VENTA)
282
283     REFERENCES VENTA(NUM_VENTA) ON UPDATE CASCADE;
284
285
286
287 ALTER TABLE VENTA_ALMACEN ADD CONSTRAINT RefARTICULO151
288
289     FOREIGN KEY (NUM_PRODUCTO)
290
291     REFERENCES ARTICULO(NUM_PRODUCTO);
292
293 CREATE VIEW TICKET AS
294     SELECT VE.RFC,VE.NUM_VENTA,VE.FECHA_VENTA, AR.NUM_PRODUCTO, VEN.PRECIO_UNITARIO,
295     VEN.CANTIDAD, VEN.PRECIO_CANTIDAD, VE.CANTIDAD_TOTAL, VE.PRECIO_TOTAL
296     FROM VENTA VE, VENTA_ALMACEN VEN, ARTICULO AR
297     WHERE VEN.NUM_VENTA = VE.NUM_VENTA
298     AND VEN.NUM_PRODUCTO = AR.NUM_PRODUCTO;
299
300 Create sequence seq_venta
301   increment by 1
302   minvalue 1
303   no maxvalue
304   start with 1
305   cache 3
306   no cycle;
307
308 drop index idx_nombres_clientes;
309 create index idx_nombres_clientes
310   on cliente(NOMBRE,APELLIDO_PATERNO,APELLIDO_MATERO);
311
312
313
314
315
```

Además se hizo una inserción de algunos registros para poderlos visualizar en la base de datos más adelante. El código SQL de los registros es el siguiente:

```

1  --insertando clientes
2  insert into CLIENTE(RFC , NOMBRE, APELLIDO_PATERNO, APELLIDO_MATERO, ESTADO, CP,
3  COLONIA, CALLE, NUMERO)
4  values ('AMMA123456789','Alejandro Martin','Avalos','Medina','Monterrey','04150',
5  'Torito','Concepcion',256);
6  insert into CLIENTE(RFC , NOMBRE, APELLIDO_PATERNO, APELLIDO_MATERO, ESTADO, CP,
7  COLONIA, CALLE, NUMERO)
8  values ('RDRC123456789','Rodrigo Daniel','Resendiz','Cruz', 'CDMX','03100',
9  'Del valle','San Borja',605);
10 insert into CLIENTE(RFC , NOMBRE, APELLIDO_PATERNO, APELLIDO_MATERO, ESTADO, CP,
11  COLONIA, CALLE, NUMERO)
12 values ('MCFA123456789','Alexis Fernando','Castro','Mora','CDMX','03400',
13  'Del valle sur','Omega',1202);
14 insert into CLIENTE(RFC , NOMBRE, APELLIDO_PATERNO, APELLIDO_MATERO, ESTADO, CP,
15  COLONIA, CALLE, NUMERO)
16 values ('RLJ0123456789','Joshua','Roldan','Landa', 'Veracruz','52501',
17  'poza rica','Mara Caibo',707);
18 insert into CLIENTE(RFC , NOMBRE, APELLIDO_PATERNO, APELLIDO_MATERO, ESTADO, CP,
19  COLONIA, CALLE, NUMERO)
20 values ('VTV0123456789','Valeria','Valdemar','Tamez','Guerrero','03100',
21  'coyula','arras',919);
22
23  --insertando emails
24  insert into CLIENTE_EMAIL(RFC,EMAIL)
25  values ('AMMA123456789','uwu@yahoo.com');
26  insert into CLIENTE_EMAIL(RFC,EMAIL)
27  values ('RDRC123456789','owo@gmail.com');
28  insert into CLIENTE_EMAIL(RFC,EMAIL)
29  values ('MCFA123456789','uwur@myspace.com');
30  insert into CLIENTE_EMAIL(RFC,EMAIL)
31  values ('RLJ0123456789','7u7@hotmail.com');
32  insert into CLIENTE_EMAIL(RFC,EMAIL)
33  values ('VTV0123456789','awa@outlook.com');
34
35  --insertando proveedores
36
37  insert into PROVEEDOR values(1,'Rosa','S.A','Guanajuato','13100','Cuajimalpa',
38  'san benito',123,'1234165371','635173567');
39  insert into PROVEEDOR values(2,'Mike','S.A','CDMX','45110','universidad',
40  'london',272,'098765432','918291');
41  insert into PROVEEDOR values(3,'Zeuz','S.A','Zacatecas','09120','Cuajimalpa',
42  'san benito',444,'9327456,null);
43
44  --insertando articulos
45
46  insert into ARTICULO values(1,'Recarga telefonica $20','Telcel',15,20,999);
47  insert into ARTICULO values(2,'Paketaxo queso 255 gr.','Sabritas',40,45.50,100);
48  insert into ARTICULO values(3,'Lapiz de grafito punta gruesa','Pelikan',5,10,45);
49  insert into ARTICULO values(4,'Paquete de papel 500 hojas tamano A40','Office
50  Depot',65,89.99,70);
51  insert into ARTICULO values(5,'coca cola light de lata 255 ml.','Coca-Cola',15,20,10);
52  insert into ARTICULO values(6,'caja de chocolates','De la Rosa',78,100,41);
53  insert into ARTICULO values(7,'impresion a color papel
54  A40','Generico',1,2.50,999);
55
56  --insertando categorias
57  insert into CATALOGO values(1,'Recarga',1);
58  insert into CATALOGO values(2,'Regalo',2);
59  insert into CATALOGO values(3,'Articulo de Papeleria',3);
60  insert into CATALOGO values(4,'Articulo de Papeleria',4);
61  insert into CATALOGO values(5,'Regalo',5);
62  insert into CATALOGO values(6,'Regalo',6);

```

```

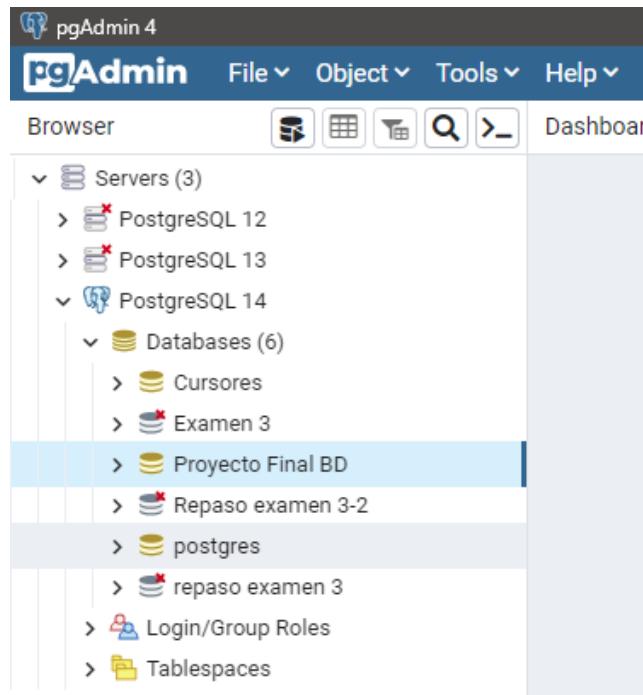
57 --insertando proveedores_almacen
58 insert into PROVEEDOR_ALMACEN values(1,1,'25-10-2021');
59 insert into PROVEEDOR_ALMACEN values(2,2,'25-10-2021');
60 insert into PROVEEDOR_ALMACEN values(2,5,'25-10-2021');
61 insert into PROVEEDOR_ALMACEN values(2,6,'25-10-2021');
62 insert into PROVEEDOR_ALMACEN values(3,3,'25-10-2021');
63 insert into PROVEEDOR_ALMACEN values(3,4,'25-10-2021');
64 insert into PROVEEDOR_ALMACEN values(3,7,'25-10-2021');
65
66 --insertando en venta
67 insert into VENTA values('VENT-1'||cast(nextval('seq_venta') as varchar),
68 '25-10-2021',65.50,2,'AMMA123456789');
69 --insertando en venta_almacen
70 insert into VENTA_ALMACEN values('VENT-1',1,20,1,(20*1));
71 insert into VENTA_ALMACEN values('VENT-1',2,45.50,1,(45.50*1));
72
73 select * from ticket;

```

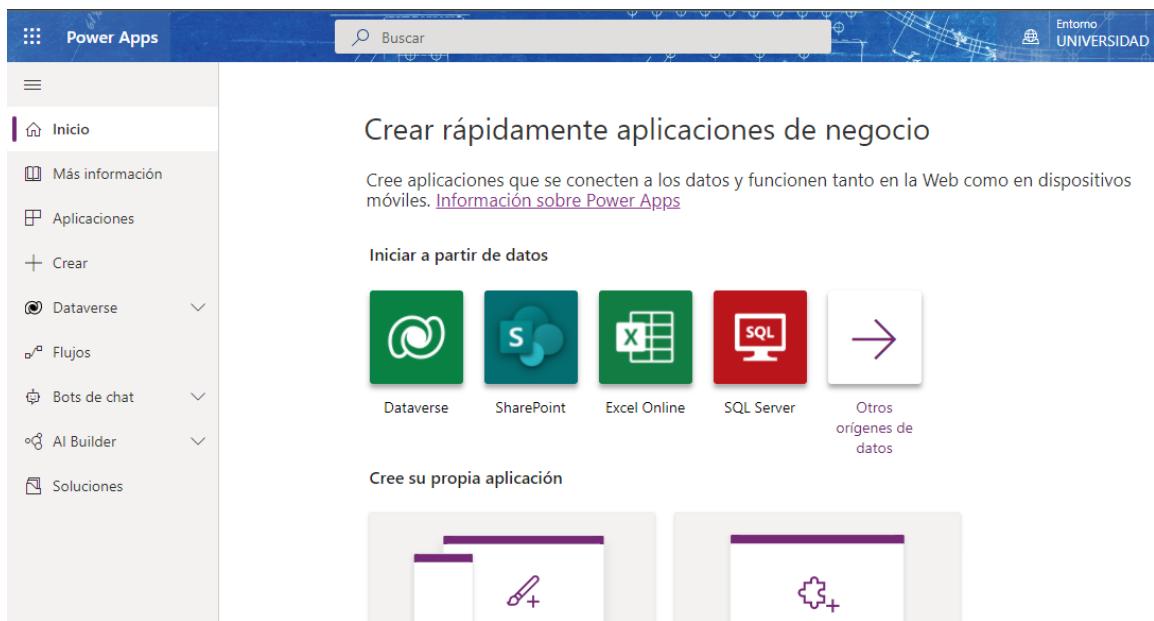
Presentación

Funcionamiento de la conexión de la base de datos a la interfaz gráfica:

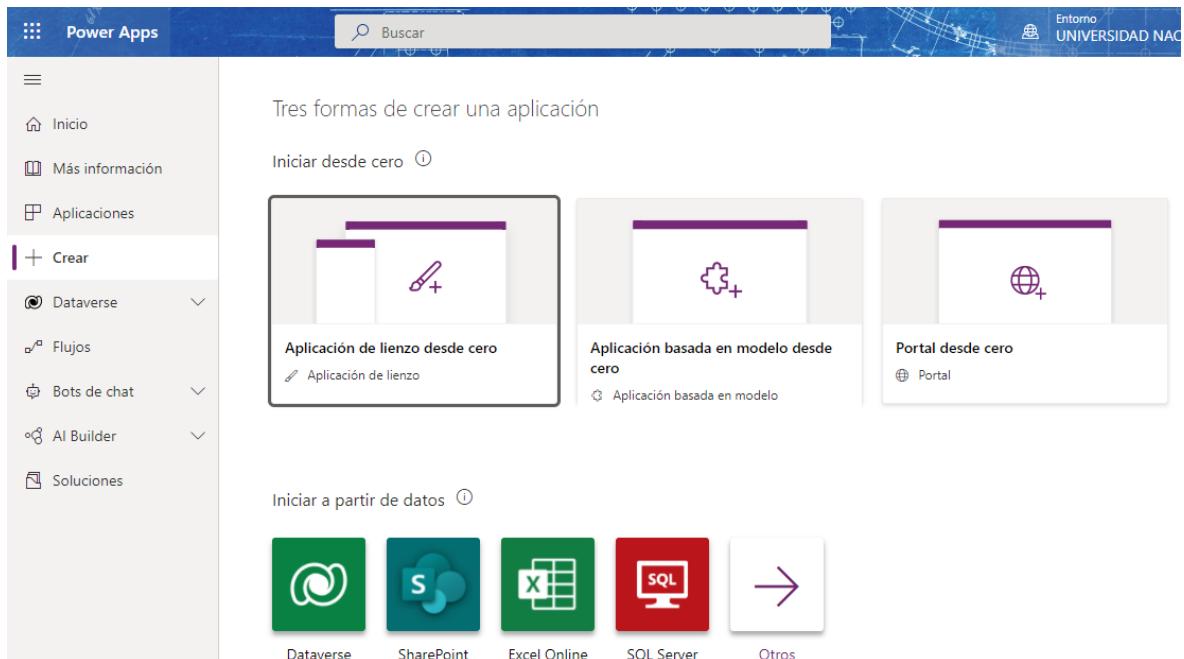
1. Tener nuestra base de datos creada en nuestro manejador, en este caso con pgAdmin 4:



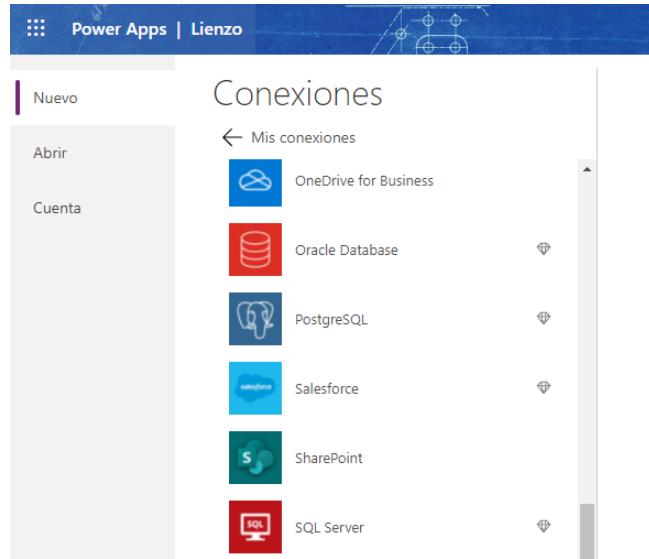
2. Tener o crear una cuenta con dominio educativo o empresarial en microsoft para acceder a una de sus aplicaciones, llamada Power Apps:



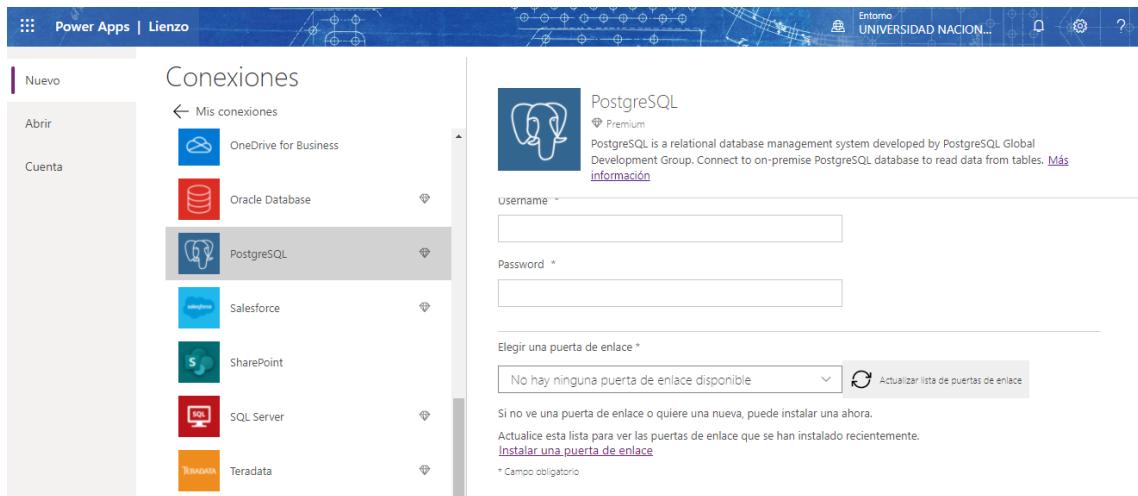
3. Despues le damos en crear y nos aparece la siguiente pantalla:



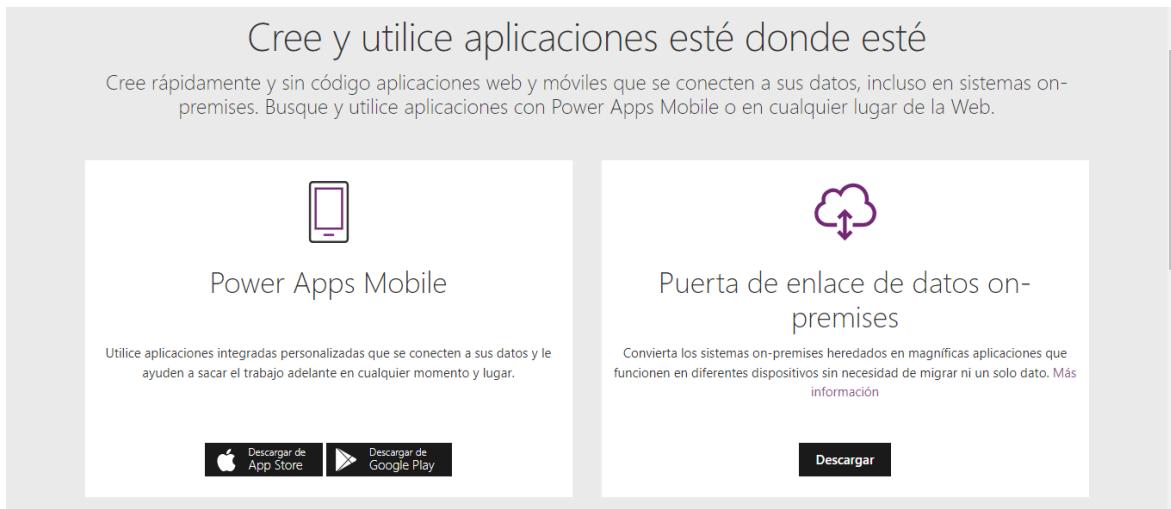
4. En la sección “Iniciar a partir de datos”, le damos clic en la flecha donde dice “Otros” y nos parece la siguiente pantalla, donde elegiremos nueva conexión y buscaremos PostgreSQL como se muestra a continuación:



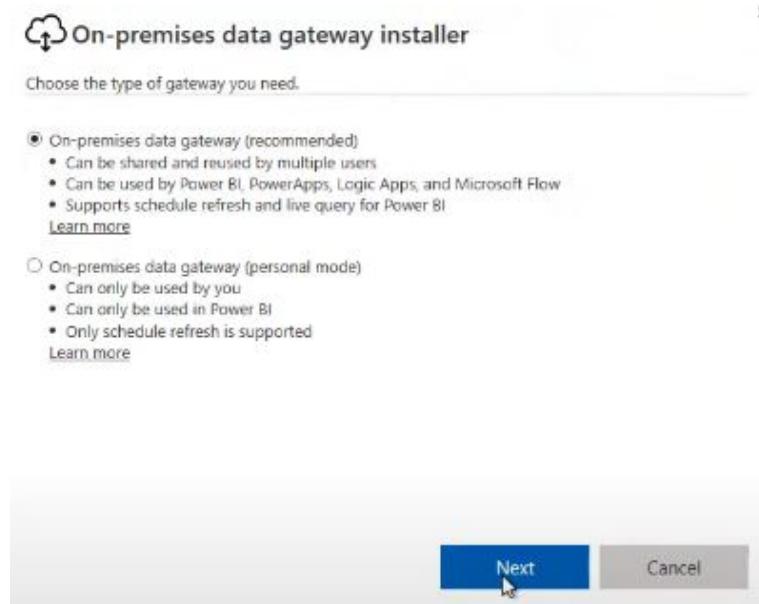
5. Seleccionamos PostgreSQL, y buscamos al final del submenú, “Instalar puerta de enlace” cómo se ve a continuación:



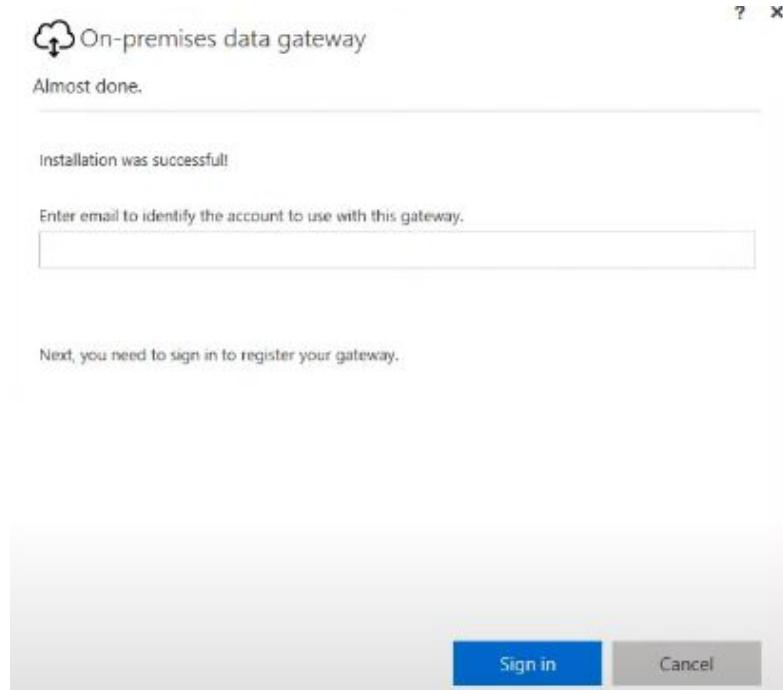
6. Despu s nos parece la siguiente pantalla, y le damos clic en “Puerta de enlace de datos on-premises” descargar:



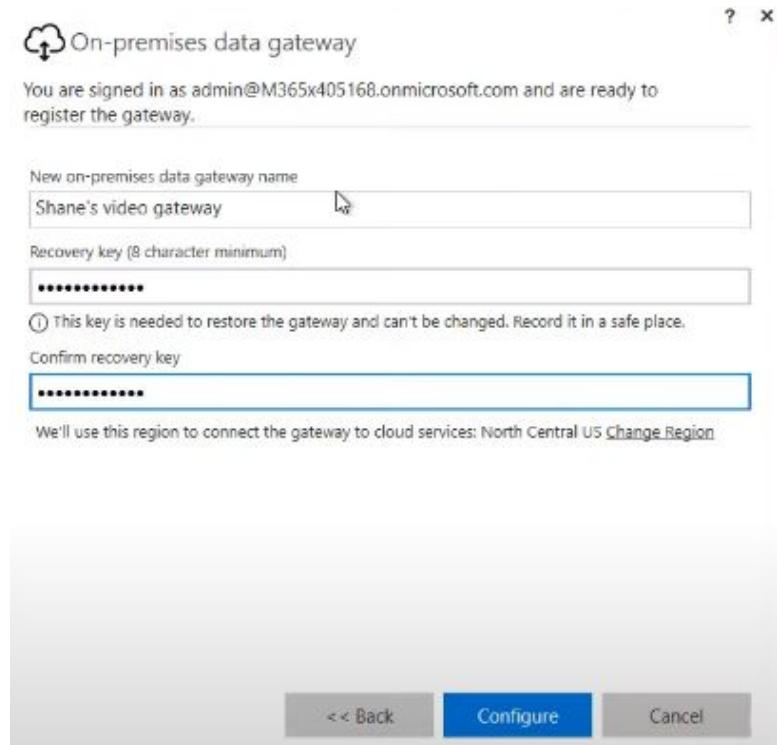
7. Despu s aceptamos los permisos y empieza a instalar como se observa en la siguiente imagen:



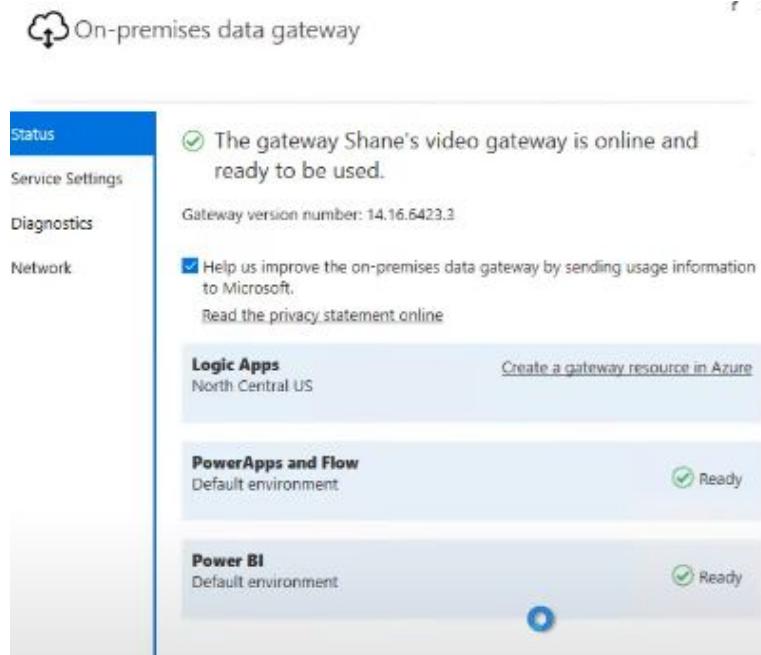
8. Luego nos pide el correo con el que iniciamos sesión en Power Apps, con el que enlazará, como se observa en la imagen:



9. En el siguiente paso nos pide el nombre de data gateway y la contraseña, cómo se observa a continuación:



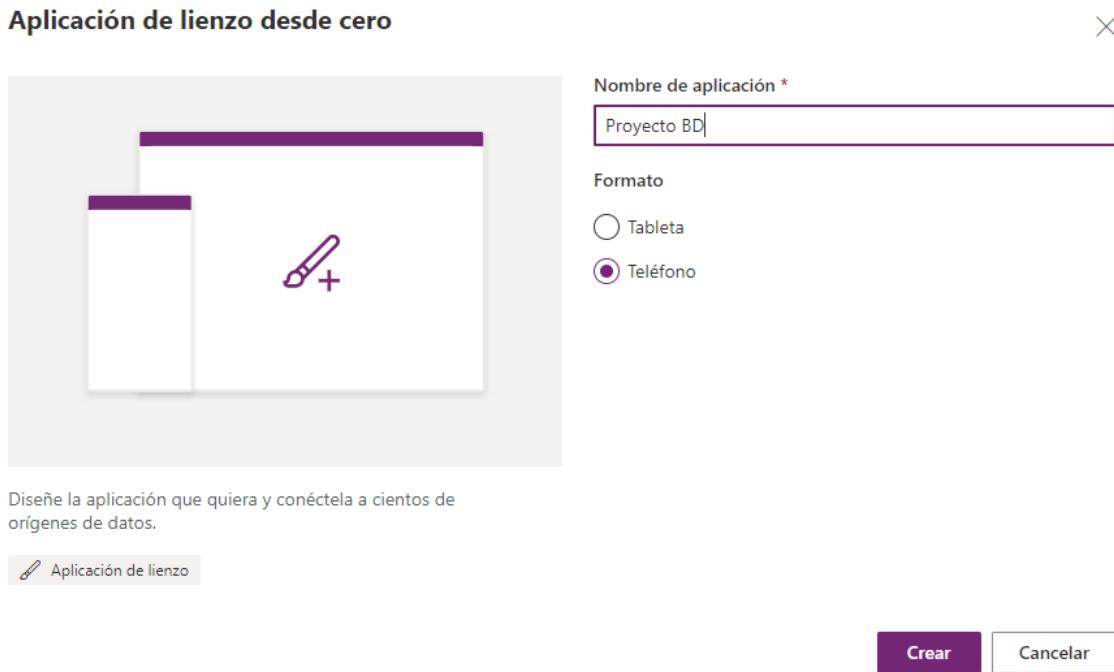
10. Despu s nos muestra el estatus del enlace con Power Apps, que fue exitoso:



11. Por  ltimo nos regresa al formulario para llenar con los datos de nuestra base de datos creada con anterioridad en el manejador pgAdmin 4:

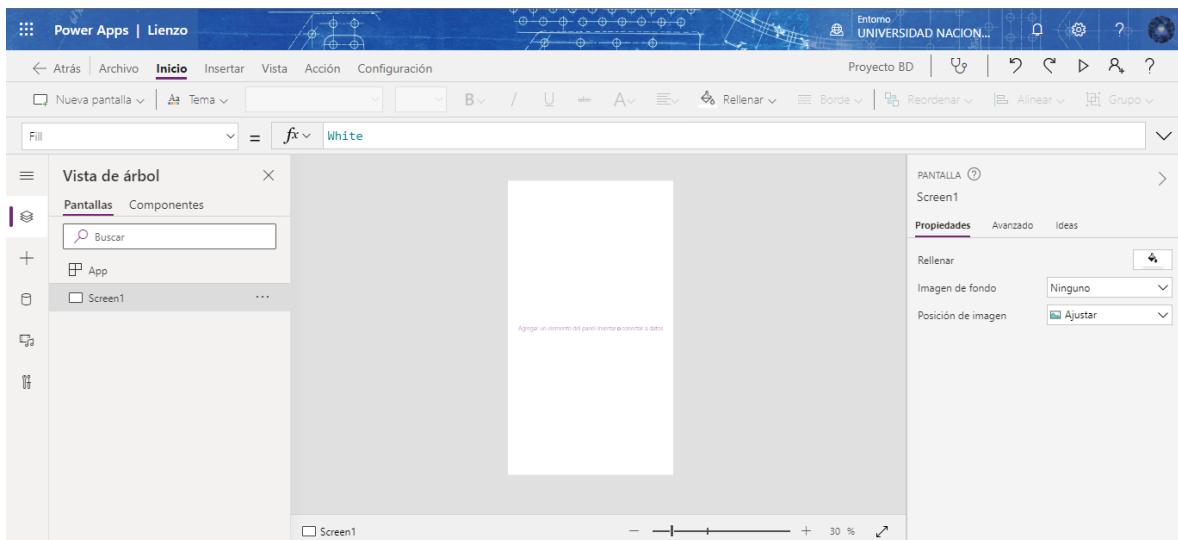
The screenshot shows the 'Conexiones' (Connections) page in pgAdmin 4. On the left, a list of available connections includes Microsoft Dataverse, MySQL, OneDrive, OneDrive for Business, Oracle Database, PostgreSQL, Salesforce, SharePoint, SQL Server, Teradata, UserVoice, and Zendesk. The 'PostgreSQL' connection is selected and expanded on the right. The configuration form for PostgreSQL includes fields for 'Server' (localhost), 'Database Name' (ProyF), 'Authentication Type' (Basic), 'Username' (postgres), 'Password' (redacted), and 'Puerta de enlace' (PostgresSQLConnect). A note at the bottom says 'Si no ve una puerta de enlace o quiere una nueva, puede instalar una ahora.'

12. Una vez ya enlazada nuestra base de datos con Power apps, nos regresamos a la página de inicio y le damos en la primera opción “Aplicación de lienzo desde cero” y nos parece la siguiente pantalla:



Donde elegimos un nombre para nuestra aplicación, en nuestro caso Proyecto BD. Formato Teléfono, y le damos en crear.

13. Nos direcciona a una nueva pantalla, donde nos permite diseñar y crear nuestra interfaz gráfica, una aplicación para smartphone, que es la siguiente:



14. Le damos clic en el símbolo de base de datos, buscamos la base de datos de PostgreSQL, en automatico Power Apps muestra la primer tabla que encuentre ya en una vista de la aplicación, como se ve a continuación:

[public].[cliente]		
Cruz	Resendiz	San Borja
Landa	Rojdan	Mora Gaito
Medina	Avalos	Concepcion
Mora	Castro	Omega
Ramirez	Priego	Sunset Vista
Tamez	Valdemar	areas

15. Por último podremos ver todas las tablas de nuestra base de datos, que podremos elegir para crear vistas o pantallas de nuestra aplicación, de acuerdo al diseño de la interfaz que queramos desarrollar, cómo observamos en la siguiente imagen:

<input type="checkbox"/> public.articulo
<input type="checkbox"/> public.catalogo
<input type="checkbox"/> public.cliente
<input type="checkbox"/> public.cliente_email
<input type="checkbox"/> public.proveedor
<input type="checkbox"/> public.proveedor_almacen
<input type="checkbox"/> public.venta
<input type="checkbox"/> public.venta_almacen
<input type="checkbox"/> Escriba el nombre de la tabla personalizada.

Funcionamiento de la aplicación:



La imagen que se muestra es la primera pantalla de la aplicación, aquí únicamente podemos observar el logo de papelería e interactuar con el botón para iniciar la aplicación.



Este es el menú principal de nuestra aplicación, como observamos, la etiqueta muestra automáticamente el nombre del usuario que esté haciendo uso de la aplicación, además de eso observamos un menú que se compone de tres opciones, de

la cuál debemos elegir una.



Esta es la pantalla de nueva venta, aquí se muestra un listado de los productos que existen en inventario junto con algunas de sus características, el usuario debe seleccionar la cantidad del producto y marcar la casilla para que el elemento se agregue al carrito, incluimos una etiqueta que hace el conteo de los elementos en el carrito, también añadimos un botón para limpiar el carrito y un ícono para regresar al menú principal. Cuando el usuario termine de escoger los productos, debe presionar el ícono del carrito.



Al presionar el ícono del carrito de compras llegaremos a esta pantalla donde tendremos un resumen de la compra, observaremos una etiqueta que nos mostrará el total de la venta, posteriormente el usuario deberá ingresar el RFC y el correo electrónico de la persona a la que se deseé enviar la orden de compra, al presionar el botón se enviará el correo y nos regresará al menú principal.

Artículos			
marca	num_producto	precio_compra	
Telcel	1	15	
Sabritas	2	40	
Pelikan	3	5	
Office Depot	4	65	
Coca-Cola	5	15	
De la Rosa	6	78	
Generico	7	1	

Proveedores			
nombre	proveedo...	razon_so...	numero
Rosa	1	S.A.	123
Mike	2	S.A.	272
Zeuz	3	S.A.	444

En el menú principal también podemos seleccionar la opción de inventario, la cual nos llevará a esta pantalla donde podremos ver la cantidad de artículos junto con sus características así como las de los proveedores, también nos podemos regresar al menú principal presionando el ícono del lado superior izquierdo.

Clientes		
<input type="text"/> Buscar en elementos		
Cruz		>
Resendiz		
San Borja		
Landa		>
Roldan		
Mara Calbo		
Medina		>
Avalos		
Concepcion		
Mora		>
Castro		
Omega		
Ramirez		>
Priego		
Sunset Vista		
Tamez		>
Valdemar		
arnas		

La tercera opción del menú principal es la de clientes, al seleccionarla nos llevará a esta pantalla donde tendremos un listado de los clientes, desde aquí podremos actualizar la base de datos, ya sea modificando, eliminando o agregando un nuevo cliente, presionando la flecha que se encuentra del lado derecho de cada registro, podemos ver los detalles del cliente.



Esta pantalla es donde se muestran los detalles de los clientes, desde aquí podremos eliminar el registro o editarlo y cual sea la opción que escojamos se verá reflejada en nuestra base de datos.

The screenshot shows a form titled "Rellene los campos" (Fill in the fields) with a list of required fields (indicated by an asterisk) and their corresponding input fields. The fields are: rfc (RDRC123456789), apellido_paterno (Resendiz), calle (San Borja), colonia (Del valle), cp (03100), estado (CDMX), nombre (Rodrigo Daniel), and numero. There are close and checkmark buttons at the top right, and a progress bar at the bottom.

Este es el menú para editar un registro existente, podremos modificar cualquier campo y al terminar los cambios debemos presionar el ícono del lado superior derecho para confirmar los cambios.

The screenshot shows a standard Windows-style dialog box titled "Rellene los campos". It contains eight text input fields, each with a required indicator (*). The fields are labeled: rfc, apellido_paterno, calle, colonia, cp, estado, nombre, and numero. The dialog has a blue header bar with the title and two icons: a close button (X) and a checkmark icon.

Esta pantalla es para agregar a un nuevo cliente, tendremos que completar los campos y al finalizar confirmar la operación presionando el ícono del lado superior derecho, así mismo podremos cancelar la operación presionando el ícono de la X.

Conclusiones

Jimenez Avila Javier Alejandro

La realización de este proyecto fue de suma importancia, ya que la problemática a resolver, se parece o es un problema de la vida real, lo cual nos generó trabajar y organizarnos cómo si fuéramos un equipo de desarrollo de una empresa. La organización de nuestros tiempos fue un punto clave para el buen desarrollo y avance del proyecto, realizamos reuniones por la plataforma meet para; revisar entregas internas, problemáticas emergentes, soluciones, y revisión de todo aquello que se hacía o faltaba. Se puso en práctica todo lo que vimos en el curso, como fue el diseño de una base de datos, por medio del desarrollo de modelos (MER, MR), la creación de tablas, llaves, consultas, operaciones en las tablas, funciones y procedimientos de una base de datos, entre otros conocimientos vistos en el curso. Además aprendimos nuevas herramientas, cómo fue la implementación de una interfaz gráfica, que de primera instancia, se nos dificulto un poco, pero encontramos

una solución sutil, por medio de Power Apps. Considero que este proyecto explotó todos los conocimientos adquiridos durante el curso, lo que reforzó y estímulo a aprender otros. Esto se vuelve una herramienta útil y de mucha importancia para nuestra formación académica.

Mora Castro Fernando Alexis

Considero que este proyecto fue un verdadero reto, sin embargo, fue muy satisfactorio ver el resultado final. Llegar al resultado final involucró aplicar todos los conocimientos vistos en clase e incluso indagar un poco más acerca de temas que no conocíamos por ejemplo como hacer la conexión de la base de datos con la interfaz gráfica, como logro personal me agració que tuve la oportunidad de aplicar conocimientos que adquirí en mi trabajo actual, pero lo más importante fue el trabajo en equipo ya que nos ayudamos mutuamente y complementamos ideas en todo momento, en conjunto pudimos elaborar desde el modelo relacional hasta la interfaz gráfica.

Resendiz Cruz Rodrigo Daniel

En este proyecto vimos todo lo que involucra hacer una base de datos desde cero, comenzando por el planteamiento de una necesidad a resolver, pasando por el diseño de la solución a manera de base de datos, la identificación de la información a guardar, la generación de las tablas y la generación de registros que ilustran el guardado de la información y terminamos con la programación de excepciones que pueden presentarse en la base de datos los cuales se representan con los trigger.

Al final pudimos ser capaces de implementar una aplicación que nos permitiera ingresar información a la base de datos. A lo largo de la realización de la base de datos logré identificar varios aciertos y errores en cuanto a nuestro desempeño, en cuanto a los aciertos puedo decir que en lo que estuvimos trabajando en el proyecto se mantuvo una buena comunicación ya que se organizaban juntas por medio de meet para tratar ciertos puntos importantes, por lo que, este tipo de comunicación dinámica nos ayudó a poder delimitar el problema y atacar ciertos puntos que no podíamos visualizar a simple vista. Por otro lado se nos dificultó iniciar con la implementación de la aplicación web ya que requería de cierto conocimiento que no teníamos, por esto mismo se tuvieron contratiempos inesperados y terminamos por optar usar la plataforma de aplicaciones power apps de microsoft para facilitar dicho desarrollo. Finalmente cabe mencionar que considero que este tipo de proyectos son el tipo de actividades que nos ayudan a poner a prueba el curso

completo ya que necesitamos de cada conocimiento adquirido a través de éste para lograrlo realizar, espero que este curso nos traiga cosas buenas a futuro ya sea si decidimos a dedicarnos a la materia o decidimos usarla como herramienta complementaria a algo a lo que nos queremos especializar.

Valdelamar Tamez Valeria

En este proyecto pudimos aprender sobre cómo realizar una base de datos para la administración de una cadena de papelerías, se realizaron procedimientos que vimos en clase y aprendimos en ella desarrollando y poniendo en práctica nuestra habilidad para programas en el lenguaje Postgres.

Finalmente se realizaron diferentes tipos de consultas para poder cumplir con los requerimientos de la base en donde podemos observar diferentes procedimientos con triggers donde podemos observar la cantidad vendida por artículo, la cantidad total que se vendió y la garantía correspondiente a la fecha, obtener el nombre de los productos de los cuales hay menos de 3 en stock así como generar una vista en la cual se tiene la información necesaria para asemejarse a una factura.