



FACULTAD DE INGENIERÍA
ASIGNATURA: BASE DE DATOS
PROYECTO FINAL

Equipo Cazainconsistencias

Pérez Gutiérrez Sandra Susana
Suxo Pérez Luis Axel

Grupo: 01

Fecha de entrega: 25 de Enero de 2020
Semestre 2021-1

Índice

1. Objetivo	3
2. Descripción del problema	4
3. Introducción	6
4. Plan de trabajo	8
5. Análisis y diseño de la base de datos	12
5.1. Modelo Entidad Relación	12
5.2. Modelo Relacional	13
5.3. Normalización	14
5.3.1. Modelo Relacional Resultante de la etapa de Diseño	19
6. Implementación	20
6.1. Implementación de modelo relacional a Base de datos	20
6.2. Creación del índice	27
6.3. Implementación de la Base de datos a una página Web	27
6.4. Programación en Javascript	30
6.5. Conexión de base de datos a partir de la página web	33
6.6. Modificaciones finales de la base de datos	36
7. Presentación	37
7.1. Conexión de la base de datos a la página web	37
7.2. Cliente servidor	39
8. Conclusiones	43
9. Referencias	45



1. Objetivo

- El alumno analizará una serie de requerimientos y propondrá una solución que atienda a los mismos, aplicando los conceptos vistos en el curso.



2. Descripción del problema

Una cadena de papelerías busca innovar la manera en que almacena su información, y los contratan para que se desarrolle 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, debería 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 porqué de la elección en ambos aspectos.

Se debe tomar en cuenta también las siguientes consideraciones:

- Puede haber distintas soluciones al problema.
- Los requerimientos enlistados anteriormente, deberán ser realizados por medio de PgSQL, con los elementos que se consideren adecuados para resolverlos.
- El número de venta debe tener un formato similar a "VENT-001", prefijo VENT, seguido de un guión y un número secuencial.
- Donde este presente el atributo domicilio, está 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.

Una vez diseñada y lista la base de datos, se debe crear una interfaz gráfica vía app móvil o web, que permita:

1. Agregar la información de un cliente.
2. Ingresar una venta, de hasta 3 artículos, los cuales podrán seleccionarse de una lista de opciones, permitir ingresar la cantidad, calcular el costo total de cada artículo y el costo total de toda la venta. Ingresar dicha información en la base de datos, respetando todas las restricciones de integridad.



3. Introducción

En este proyecto busca solucionar un problema que podemos enfrentarnos en la vida laboral, el proyecto busca en general que se apliquen los conceptos básicos de la materia de base de datos vistos a lo largo del curso, y su finalidad es que al concluir el proyecto se deben mostrar las habilidades adquiridas a lo largo del curso, principalmente trata de poner a prueba los conocimientos adquiridos en el curso, las habilidades teóricas y prácticas, adicional el poder conocer o reafirmar el uso de herramientas que se pueden brindar a un consumidor final, para lo cual como se ve en la descripción el proyecto el problema a resolver es el crear un sistema para una papelería, con el objetivo de que el sistema se gestione mediante una base de datos.

Durante el desarrollo del proyecto Se realizaran simulaciones con el fin de poder verificar el funcionamiento de la base de datos, principalmente con la finalidad de tratar de encontrar posibles errores o fallas previo a la presentación final para así en caso de tener algún detalle o encontrar algún tipo de error o mal funcionamiento, se pueda documentar a la par que se le buscará posibles soluciones, las cuales igualmente para fines de aprendizaje quedarán documentadas en el documento así como al manera en que se logra solucionar, adicional se desarrollará un pequeño manual explicando brevemente las funcione y acciones que se van a poder realizar y la forma de hacerlo.

Pasando a la parte de diseño, se buscará lograr el mejor análisis posible para así al momento de diseñar el modelo entidad relación, el cual mediante la documentación se buscará que quede claro la razón de ser del diseño final y las modificaciones oportunas que se hicieron durante el desarrollo, donde se explicaran el motivo de ser de cada clave primaria, las cuales serán las llaves primarias en cada tabla, así como las llaves foráneas esto con la finalidad que cualquier persona ajena al área de informática o con pocos conocimientos o nulos sobre las bases de datos pueda entender las funciones de la misma, con lo cual en caso de ser necesario se incluirán glosarios con la finalidad de dejar en claro los tecnicismos que se puedan presentar para que sean más claros para personas que desconozcan del tema.

Conforme se vaya profundizando en el desarrollo del diseño del proyecto se verán las formas de normalización, indicando la llave primaria, y también cada etapa de normalización, con la meta de llegar hasta la tercera forma normal y en caso de que se conveniente usar una cuarta forma normal o alguna extra se dará una justificación de la decisión, de igual forma si en el diseño se requiriera por cuestiones definidas que las normalizaciones realizadas no se apliquen y se decida no utilizar esa forma normal.

En el diseño de las tablas, específicamente en los atributos se definirá el tipo de dato y junto con el tipo de dato las propiedades que estos tendrán y si es necesario usar constraint se detallarán las razones de ser de los mismos, para nosotros lo más importante es que la base de datos cumpla con los requerimientos planteados por el planteamiento del problema, y es por eso que la plan de trabajo y el diseño son partes importantes para lograrlo, con el mejor resultado.

Para justificar la importancia de las bases de datos, y lo importante que es el diseño de esta al momento de realizar consultas y/o un ingreso de información, esto de forma rápida y en buscando que la información almacenada esté segura dando al cliente la confianza que además de tener un diseño óptimo en relación a la funcionalidad se tengan todas las medidas para evitar que la información



recabada se utilice de formas no adecuadas, o por personas no autorizadas, además de mantener la información accesible para cada usuario según tenga los permisos otorgados.

En el proyecto se requiere que se combine con conocimientos de otras materias, como Ingeniería de Software, Administración de Proyectos de software, además de todas las materias relacionadas directa o indirectamente con programación estructurada y orientada a objetos, ya que como parte del proyecto es necesario, aplicar estos conocimientos para desarrollar la parte de aplicación.



4. Plan de trabajo

Se usó el software de monday para programar las tareas y reuniones a lo largo del proceso de desarrollo de software, la finalidad de esto es mantener un control adecuado de las tareas, así como tener una fácil visualización de las tareas asignadas, su progreso y su prioridad, ya que esta herramienta facilita visualizar todos estos aspectos.

En general se organizó el proyecto por módulos o secciones con el fin de llevar mayor control sobre áreas específicas y así definir de una forma mejor las tareas que se llevarán a cabo, los módulos que se consideraron son los siguientes:

- Conexión de Base de Datos: en esta sección se busca definir los puntos específicos relacionados con la conexión de la base de datos y la página web. Las tareas en específico son: elección de roles, investigar sobre la conexión de base de datos, conexión con el carrito, realizar pruebas de fluidez en las conexiones.
- Desarrollo de Base de datos: en esta sección se abarcan las tareas relacionadas con el análisis, diseño y programación de la base de datos , en donde las tareas específicas son: análisis y diseño del modelo entidad relación, análisis y diseño del modelo relacional, normalización, realización de scripts, hacer pruebas con la base de datos, realizar correcciones a las bases de datos, hacer pruebas de inserción de datos, realizar la conexión de la base de datos, realización de los queries que se requieren para los requerimientos específicos del manejador de datos.
- Página Web: en esta sección se designarán las tareas relacionadas con el diseño de la página web, así como su desarrollo, la principal tarea consiste en el diseño y desarrollo de la página web.
- Documento: finalmente esta sección tiene que ver con el documento formal escrito, así como algunos entregables que no estén considerados en los puntos anteriores, como parte de las tareas de este punto se encuentran: el diseño del documento, el documento escrito formal que se entregará, la presentación y el desarrollo de un manual básico para el usuario.

Debido a los periodos de prueba del monday es necesario sacar una nueva cuenta, para poder actualizar la información de los avances del proyecto, por lo cual para fines del documento escrito y tener una visión general de la evolución del proyecto, se dejarán algunas capturas adicionales de cómo fue evolucionando el proyecto, cuya finalidad es poder tener una visión de cómo se fueron designando las tareas, así como, la evolución y términos de esta.

Primer corte de tareas:



Conexión de base con página	Responsable	Estado	Prioridad	Inicio	Tiempo Estimado	Término	Estado
Elegir roles de base de datos, servidor y cliente	SS AM	Listo	Alta	dic. 15	1h	dic. 15	Done
Buscar sobre conexión de datos a la página	SS AM	En proceso	Alta	dic. 15	8h	dic. 22	Working on it
					9h Total	dic. 15 - 22	
Desarrollo de Base de datos	Responsable	Estado	Prioridad	Inicio	Tiempo Estimado	Término	Estado
Modelo Entidad Relación	SS	Listo	Alta	dic. 6	3h	dic. 13	Done
Modelo Relacional	SS	Listo	Alta	dic. 6	3h	dic. 13	Done
Diseño del documento	SS	Listo	Media	dic. 6	3h	dic. 13	Done
Normalización	SS	Listo	Media	dic. 13	3h	dic. 14	Done
Hacer los scripts correspondientes a las b...	SS	Listo	Alta	dic. 15	8h	dic. 22	Done
Probar la base de datos	SS	En proceso	Alta	dic. 18	8h	dic. 22	Working on it
					28h Total	dic. 13 - 22	

Segundo corte de tareas:

Conexión de Base de Datos	Propietario	Estado	Prioridad	Fecha de Inicio	Horas invertidas	Fecha de Cierre
Elegir roles		Listo	Prioridad Baja	Dec 15, 2020	1 h	Dec 15, 2020
Buscar sobre la conexión de la base de datos		Listo	Prioridad Media	Dec 15, 2020	4 h	Dec 21, 2020
Probar fluidez de base de datos		Trabajando en ello	Prioridad Media			
Carrito		Trabajando en ello	Prioridad Media			
+ Add Task						
					Dec 15	5 h sum
						Dec 15 - 21

Desarrollo de Base de datos	Propietario	Estado	Prioridad	Fecha de Inicio	Horas invertidas	Fecha de Cierre
Modelo Entidad Relación		Listo	Prioridad Alta	Dec 6, 2020	3 h	Dec 13, 2020
Modelo Relacional		Listo	Prioridad Alta	Dec 6, 2020	3 h	Dec 13, 2020
Normalización		Listo	Prioridad Media	Dec 13, 2020	3 h	Dec 14, 2020
Hacer los scripts correspondientes		Listo	Prioridad Alta	Dec 15, 2020	8 h	Dec 22, 2020
Probar la base de datos		Trabajando en ello	Prioridad Alta	Dec 18, 2020	8 h	Dec 22, 2020
Corregir la base de datos		Listo	Prioridad Alta	Dec 21, 2020	3 h	Dec 22, 2020
Inserción de datos		Trabajando en ello	Prioridad Media			
Conexión de Base de datos		Trabajando en ello	Prioridad Media	Dec 22, 2020	3 h	Dec 28, 2020
Queries de los requerimientos en la base de dat...		Trabajando en ello	Prioridad Media			
+ Add Task						
					Dec 6 - 22	31 h sum
						Dec 13 - 28



Página Web	Propietario	Estado	Prioridad	Fecha de Inicio	Horas invertidas	Fecha de Cierre
Diseño y desarrollo de la página web		Trabajando en ello	Prioridad Alta	Dec 6, 2020	12 h	Dec 28, 2020
+ Add Task						
				Dec 6	12 h sum	Dec 28
Documento	Propietario	Estado	Prioridad	Fecha de Inicio	Horas invertidas	Fecha de Cierre
Diseño del documento		Listo	Prioridad Media	Dec 6, 2020	3 h	Dec 13, 2020
Documento Escrito		Trabajando en ello	Prioridad Media	Dec 6, 2020		Jan 17
Presentación		Trabajando en ello	Prioridad Baja	Dec 30, 2020		Jan 17
+ Add Task						
				Dec 6 - 30	3 h sum	Dec 13 - Jan 17

Una vez definidas las tareas, se procedió a repartirlas entre el equipo de trabajo, de tal manera de buscar que las actividades se repartirán de la mejor manera y se busca aprovechar los conocimientos que cada uno de los integrantes domina.

Además, que con el apoyo de monday se logró un llevar un mejor orden y control de las tareas asignadas, así como visualizar las tareas que requieren mayor prioridad, adicional con monday nos podemos apoyar en agregar tareas que se vayan requiriendo conforme avanza el proyecto, con lo cual es fácil gestionar las tareas que se tienen previstas, así como incluir nuevas tareas con facilidad.

Otro de los beneficios en monday es que este tiene facilidad para agendar y organizar reuniones, ya que la herramienta permite llevar a modo de agenda las reuniones previstas para el desarrollo del proyecto.

En las siguientes imágenes se mostrará la evolución de las reuniones realizadas con el fin de designar tareas nuevas o ver avances de las diversas tareas asignadas al proyecto.

Primer corte de reuniones:

Reuniones	Persona	Estado	Fecha	Observaciones
Reunión 6-12-2020		Listo	dic. 8	
Reunión 13-12-2020		Listo	dic. 13	
Reunión 14-12-2020		Listo	dic. 14	
Reunión 15-12-2020		Listo	dic. 15	

Segundo corte de reuniones:



Reuniones

	Date
Reunión	Dec 6, 2020
Reunión	Dec 13, 2020
Reunión	Dec 14, 2020
Reunión	Dec 20, 2020
Reunión con el profesor para presentar avances y plantear dudas	Dec 21, 2020
Reunión	Dec 22, 2020
Reunión	Dec 30, 2020
Reunión	Jan 3
+ Add	

Finalmente este es el diagrama de Grantt con todas las realizadas:





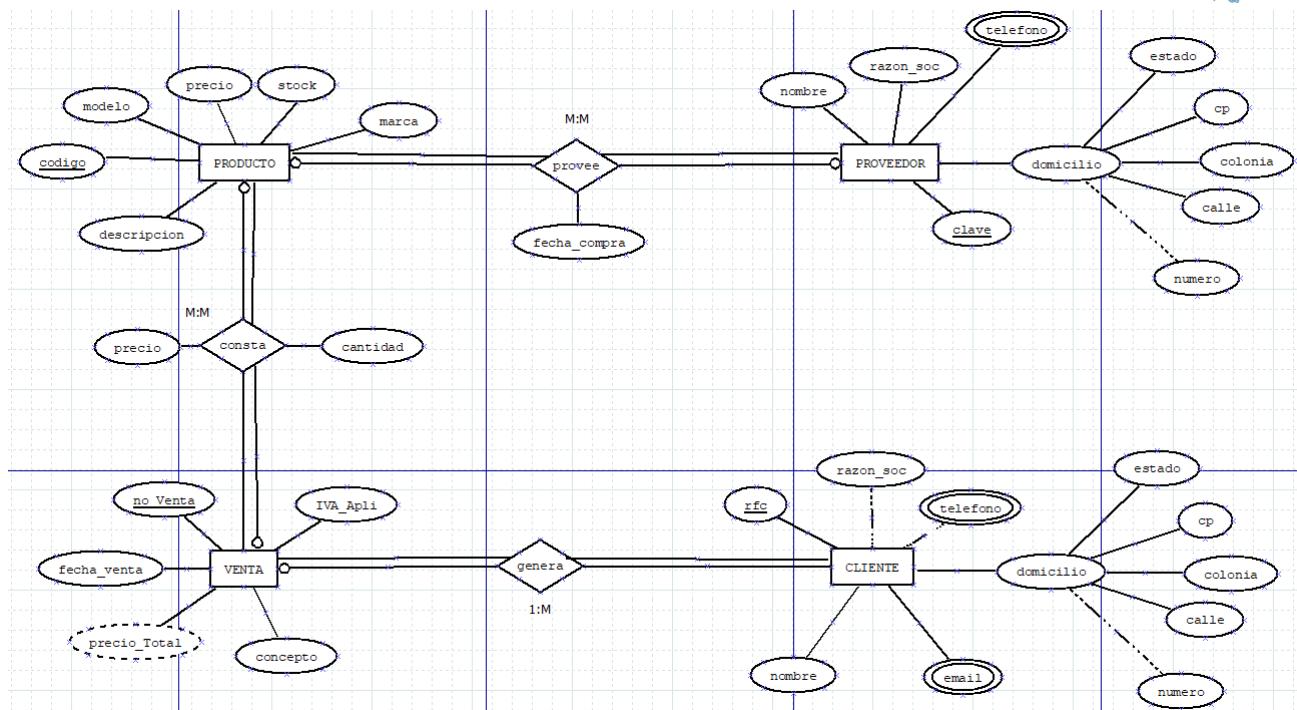
5. Análisis y diseño de la base de datos

5.1. Modelo Entidad Relación

Abstrayendo el problema planteado en el **Inciso 2** se obtuvieron las siguientes entidades:

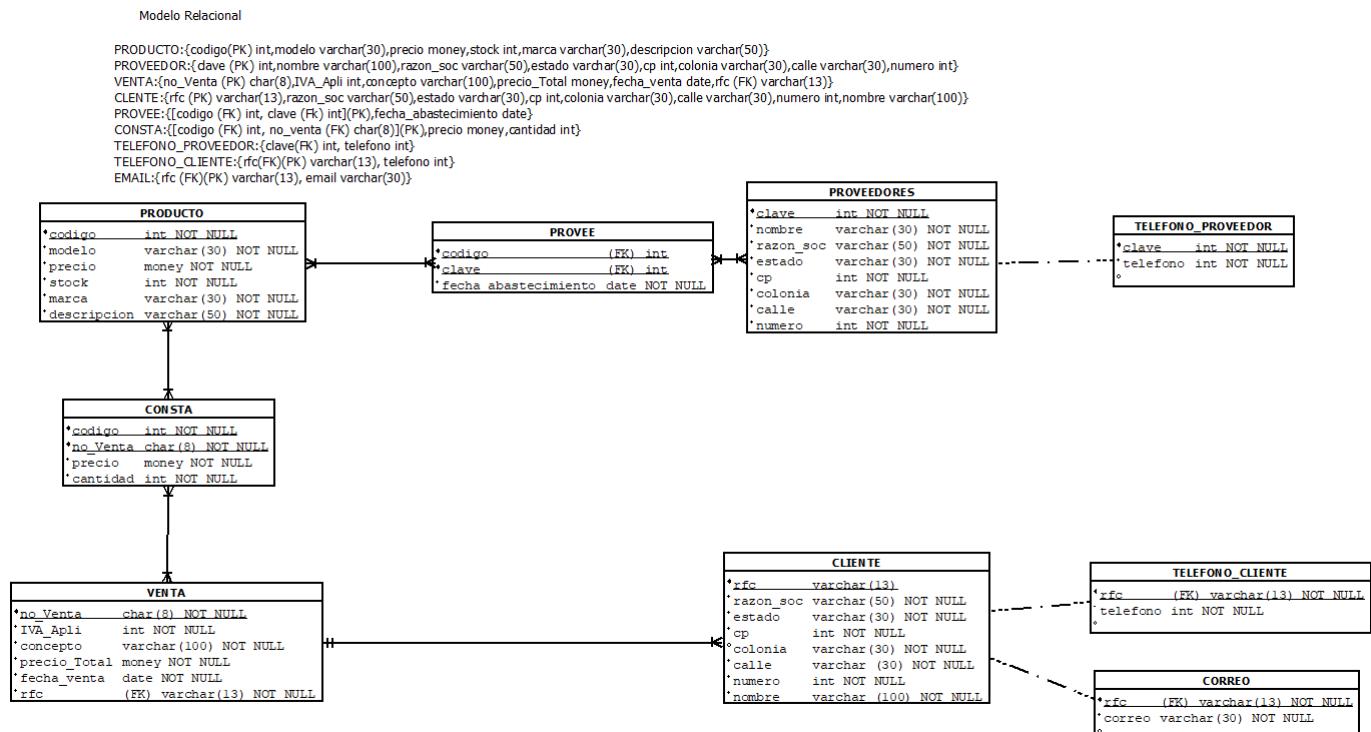
- Producto: Esta entidad se refiere a los productos que contiene la papelería, los cuales son abastecidos por un proveedor y participan en la venta por medio de la acción de comprar o vender, de esta entidad lo importante es tener en cuenta los atributos del producto que son: el código del producto, marca, modelo, descripción, stock y precio.
- Venta: Esta entidad guarda todo lo relacionado con la venta en sí. Con esto se refiere al número de venta, precio total, IVA, fecha de venta, concepto de venta, cuya finalidad es poder ver los artículos vendidos.
- Cliente: Esta entidad se refiere a los datos personales de cliente, y el RFC lo tomamos como clave primaria por el requisito para hacer facturas automáticas y a la vez por las características de este atributo es ideal para que se defina como llave primaria, adicional al requerirse que se guarde el RFC para la generación de facturas de manera automática refuerza el hecho que debido a las características previamente comentadas se haya optado por usarla como clave primaria. Los atributos multivaluados que consideramos son teléfono y e-mail, porque se puede dar el caso que el cliente tenga varios teléfonos y/o correos.
- Proveedor: Esta entidad se refiere a los datos de los proveedores de la papelería, la clave primaria es artificial ya que fue propuesta por nosotros los diseñadores debido a que no se tiene un atributo de los definidos en el planteamiento del problema que se pueda considerar único. El atributo multivaluado que se considera es teléfono, debido a que el cliente puede tener varios teléfonos.

A continuación, se muestra el diagrama del modelo entidad relación obtenido debido al análisis previo del cual se partirá para obtener el modelo entidad relación.



5.2. Modelo Relacional

Partiendo del modelo Entidad Relación se desarrolló el modelo Relacional correspondiente, para esto se aplican los conocimientos teóricos vistos en clase.





5.3. Normalización

Para la normalización se simuló el llenado de datos en una tabla para así normalizar la estructura en caso que esto sea requerido, además de poder ver de manera visual en una tabla todos los datos el tipo de normalización que se va requiriendo según sea el caso.

PRODUCTO					
codigo	modelo	precio	stock	marca	descripción
1122	Premium	40	1	Bic	Utilizado para la escritura de grabados en discos
2233	Premium	20	9	Norma	Utilizado para la limpieza de discos
3344	Basico	15	9	Norma	Utilizado para la limpieza de discos
4455	Basico	7	7	Bic	Utilizado para la escritura de grabados en discos

Se analizará cada una de las entidades, que generan una tabla, con el fin de tener claridad para cada una de los procesos de normalización, para la tabla (antes entidad) PRODUCTO:

¿La tabla previa se encuentra en 1FN?

Si debido a que en 1FN no existen grupos de repetición.

¿La tabla previa se encuentra en 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

PROVEE		
codigo	clave	fecha_compra
1122	9999	04/12/2014
2233	9998	06/09/2018
3344	9997	06/09/2018
4455	9996	25/12/2019

Ahora al observar la tabla PROVEE se tiene lo siguiente:

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.



CONSTA			
codigo	no_Venta	precio	cantidad
1122	VENT-001	40	1
2233	VENT-002	40	2
3344	VENT-003	45	3
4455	VENT-004	28	4

Ahora se procede a revisar la tabla CONSTA:

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

VENTA					
no_Venta	IVA_Apli	concepto	precio_Total	fecha_venta	rfc
VENT-001	16	PUEDE	46.4	13/12/2020	SUPL970118
VENT-002	16	QUEDAR	46.4	13/12/2020	PGSS980611
VENT-003	16	COMO	52.2	18/01/2018	MDCA920711
VENT-004	16	ATRIBUTONULL	32.48	21/12/2019	GATO940511

Ahora repetimos el proceso para la tabla VENTA, y así poder definir si esta ya esta en 3FN o necesita llevar acabo el proceso de normalización para lo cual se realizan las siguientes preguntas:

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

PROVEEDORES							
clave	nombre	razon_soc	estado	cp	colonia	calle	numero
9999	konami	PUEDE	CDMX	4700	dregal de carras	Rin. de los juegos	4
9998	microsoft	SER	Tabasco	2547	Paraiso	Pijul	4
9997	DELL	NULO	CDMX	7894	eroes de padiem	Cancun	147
9996	Huawei		CDMX	4700	dregal de carras	Rin. monedas	204



Finalmente se analiza la última tabla la cual se llama PROVEEDORES, como sigue a continuación:
¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

No se encuentra en 2FN porque hay dependencias parciales. Por tanto se normalizó como se muestra a continuación:

PROVEEDORES						
clave	nombre	razon_soc	cp	colonia	calle	numero
9999	konami	PUEDE	4700	Pedregal de carrasco	Rin. de los juegos	4
9998	microsoft	SER	2547	Paraíso	Pijul	4
9997	DELL	NULO	7894	Heroes de padriena	Cancun	147
9996	Huawei		4700	Pedregal de carrasco	Rin. monedas	204

COLONIA	
cp	estado
4700	CDMX
2547	Tabasco
7894	CDMX

Pasando el proceso de normalización se procede a revisar detenidamente si se requiere una normalización adicional o esta ya se encuentra en 3FN por lo cual al analizarla se tiene lo que sigue.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

CLIENTE							
rfc	nombre	razon_soc	estado	cp	colonia	calle	numero
SUPL970118	Luis Axel Suxo Pérez	PUEDE	CDMX	4700	Pedregal de carrasco	Rin. de los juegos	4
PGSS980611	Pérez Gutiérrez Sandra Susana	SER	Tabasco	2547	Portales	cristales	4
MDCA920711	Moreno del Castillo Araceli	NULO	CDMX	7894	Aeropuerto	puerta 11	147
GATO940511	García Tista Edgar		CDMX	4700	Copilco	pumas	204

Se continua el proceso con la tabla cliente, la cual se analizará y se determinará el proceso que se debe llevar con la misma, como se muestra a continuación:

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

No se encuentra en 2FN porque hay dependencias parciales. Por tanto se normalizó como se muestra en las tablas siguientes, ya que fue necesario que durante el proceso de normalización la tabla inicial se convierta en dos tablas:



COLONIA	
cp	estado
4700	CDMX
2547	Tabasco
7894	CDMX

CLIENTE						
rfc	nombre	razon_soc	cp	colonia	calle	numero
SUPL970118	Luis Axel Suxo Pérez	PUEDE	4700	Pedregal de carrasco	Rin. de los juegos	4
PGSS980611	Pérez Gutiérrez Sandra Susana	SER	2547	Portales	cristales	4
MDCA920711	Moreno del Castillo Araceli	NULO	7894	Aeropuerto	puerta 11	147
GATO940511	García Tista Edgar		4700	Copilco	pumas	204

NOTA: Se aprovechó la tabla de la normalización de proveedores para normalizar esta tabla.

Ahora se procede a revisar si cumple con la 3FN.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

TELEFONO_PROVEEDOR	
Clave	Teléfono
1234	5532286547
5678	5521175478
1478	5522887799
3698	5513792468

Continuando con la revisión de tablas y si estas estan normalizadas, toca el turno de la tabla TELEFONO PROVEEDOR.

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.



TELEFONO_CLIENTE	
rfc	Teléfono
SUPL970118	5532286547
PGSS980611	5530326816
MDCA920711	5512997828
GATO940511	5513793214

La siguiente tabla que se analiza es la tabla TELEFONO CLIENTE.

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.

CORREO	
rfc	correo
SUPL970118	axelsuxo@hotmail.com
PGSS980611	sandraconnors271198@gmail.com
MDCA920711	xtravaganzza.17@gmail.com
GATO940511	gatoveloz@gmail.com

Finalmente para la ultima tabla denominada CORREO, se procede a revisar si se encuentra o no normalizada, con lo que se tiene lo siguiente:

¿La tabla previa se encuentra 1FN?

Se encuentra en 1FN porque no hay grupos de repetición.

¿La tabla previa se encuentra 2FN?

Se encuentra en 2FN porque no hay dependencias parciales, todos los atributos dependen de la clave primaria.

¿La tabla previa se encuentra 3FN?

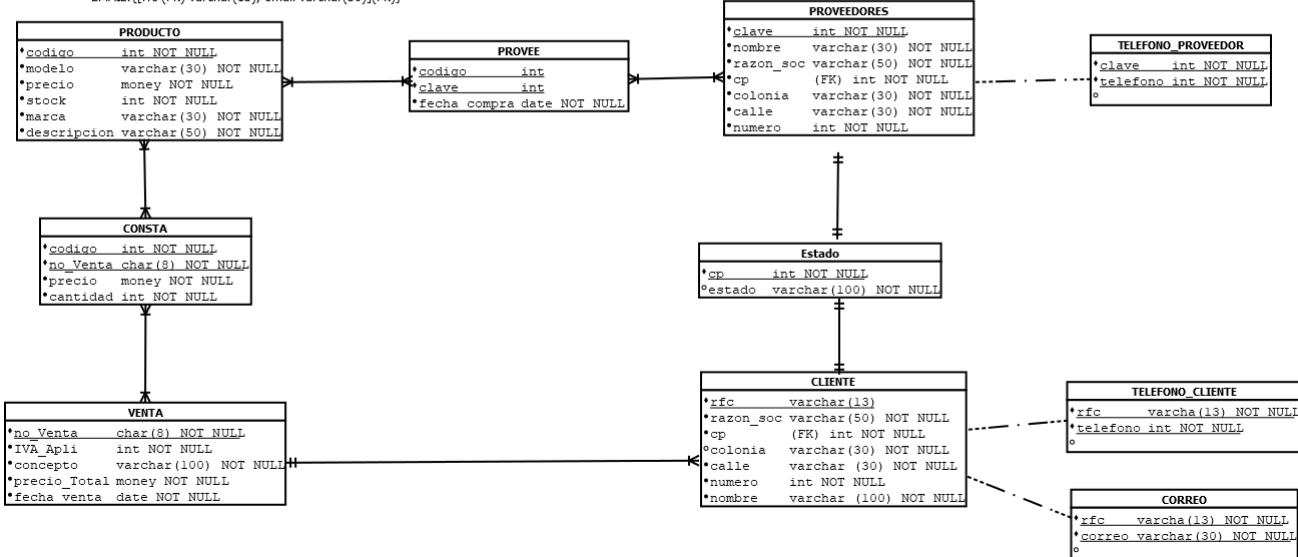
Se encuentra en 3FN porque no hay atributos que tengan una dependencia entre otro atributo que no sea llave primaria o candidata.



5.3.1. Modelo Relacional Resultante de la etapa de Diseño

Modelo Relacional

```
PRODUCTO:{codigo(PK) int,modelo varchar(30),precio money,stock int,marca varchar(30),descripcion varchar(50)}  
PROVEEDOR:{clave (PK) int,nombre varchar(100),razon_soc varchar(50),estado varchar(30),cp int,colonia varchar(30),calle varchar(30),numero int}  
VENTA:{no_Venta (PK) char(8),IVA_Apli int,concepto varchar(100),precio_Total money,fecha_venta date}  
CLENTE:{rfc (PK) varchar(13),razon_soc varchar(50),estado varchar(30),cp int,colonia varchar(30),calle varchar(30),numero int,nombre varchar(100)}  
PROVEE:{codigo (FK) int, clave (Fk) int,(PK),fecha_compra date}  
CONSTA:{[codigo(FK) int, no_venta (FK) char(8)][PK],precio money,cantidad int}  
TELEFONO_PROVEEDOR:{[clave(FK) int, telefono int][PK]}  
TELEFONO_CLIENTE:{[rfc(FK) varchar(13), telefono int][PK]}  
EMAIL:{[rfc (FK) varchar(13), email varchar(30)][PK]}
```





6. Implementación

Primero para esta parte y como parte de la planeación se asignaron los roles establecidos para poder realizar las conexiones, así como tener la máquina cliente, la máquina servidor y donde se hospedará la base de datos, con la finalidad de evitar saturar los equipos de cómputo de los participantes y tener una respuesta ágil dados los recursos.

Para realizar las conexiones se tenía pensado hacer los siguientes roles:

- Servidor: Sandra Susana Pérez Gutiérrez.
- Cliente: Suxo Pérez Luis Axel.

A causa de no tener la ip homologada no se pudo realizar de forma remota como lo teníamos pensado, pero seguiremos usando la implementación de cliente servidor, pero con ayuda de máquinas virtuales se realizaran estas conexiones, esto también suele verse en algunos lugares para realizar pruebas en ambientes controlados, principalmente para las pruebas de conexión.

Otra solución sería contratar el servicio de internet de otro sitio que no sea Izzi o realizar las solicitudes para nos proporcionen la ip homologada, con Telmex hacer este tipo de conexiones no presenta ningún inconveniente porque ellos manejan la ip pública de forma distinta, es por eso que es importante al momento de contratar saber qué tipo de ip pública proporciona el servicio.

6.1. Implementación de modelo relacional a Base de datos

Como parte de la implementación y para fines de este documento se solicitó lo siguiente:

- Al recibir el código de barras de un producto, regrese la utilidad.

Para poner una solución a este punto importante del proyecto se implementó una función que recibiera el código de barras y retornara la utilidad del producto.

Para esto primero se analiza la tabla producto para identificar cuáles son las columnas de código y utilidad (columna la cual nosotros le llamamos precio).



```
1 SELECT * FROM public.producto
2 ORDER BY codigo ASC
```

Data Output Query History Explain Messages Notifications

	codigo [PK] integer	modelo character varying (30)	precio money	marca character varying (30)	descripcion character varying (150)	stock integer	nombre character varying (50)	imagen character varying (100)	categoria character varying (100)
1	1	Básico	46,50 €	BIC	Pluma de tinta negra y punto ...	1	Paquete pluma y lápices Bic	assets/images/17875-515x5...	Escolares
2	2	Profesional	44,97 €	Scribe	Cuaderno de cuadro chico de ...	24	Cuaderno Profesional (cuadr...	assets/images/1-500x500.jpg	Escolares
3	3	Básico	46,50 €	BIC	Lápiz clásico turquesa del nú...	3	Lápiz Bic	assets/images/2-676x676.jpg	Escolares
4	4	Básico	17,00 €	Pelikan	Goma rectangular en un emp...	4	Goma de Migajón PELIKAN ...	assets/images/1-3-1046x104...	Escolares
5	5	Profesional	44,97 €	Scribe	Cuaderno de cuadro chico de ...	5	Cuaderno Profesional (cuadr...	assets/images/1-500x500.jpg	Escolares
6	6	Profesional	2.312,00 €	Koi	Estuche con 60 pastillas de a...	6	Acuarela	assets/images/3-500x500.jpg	Artes
7	7	Profesional	44,97 €	MOLOTOW	Cuaderno de 39X27 cm con 2...	7	Cuaderno de dibujo	assets/images/4-500x500.jpg	Artes
8	8	Profesional	46,50 €	Blanca nieves	Lápiz de color profesional de ...	8	Lápices de colores	assets/images/estuche-lapic...	Artes
9	9	Profesional	438,90 €	Pinto	Estuche de pinceles de nylon ...	9	Pinceles profesionales	assets/images/218590-500x...	Artes
10	10	Profesional	70,97 €	Ati	Godete desechable de plástic...	10	Godete	assets/images/436-500x500.j...	Artes
11	11	Profesional	572,18 €	Marc	Paquete de 60 rotuladores de ...	11	Rotuladores de colores mar...	assets/images/rotulador.jpg	Artes
12	12	Profesional	780,00 €	SPEEDBALL	Este conjunto de herramienta...	12	GUBIAS SPEEDBALL 4131 In...	assets/images/372.jpg	Artes
13	13	Profesional	5.190,00 €	REMBRANDT	Set de iniciación Rembrandt c...	13	Colores pastel REMBRANDT	assets/images/pasteles.jpg	Artes
14	14	Profesional	126,00 €	Moxy	Botella de 4.23 fl oz de esmal...	14	Pegamento con brillantina a...	assets/images/pega.jpg	Artes
15	15	Profesional	539,20 €	bostitch	Engrapadora de golpe para us...	15	Engrapadora de golpe bostit...	assets/images/engrapadora.j...	Oficina
16	16	Profesional	399,00 €	Art Deco	Libro de arte decó con los má...	16	Jumbo Slim Art Deco	assets/images/libro.jpg	Artes
17	17	Profesional	649,00 €	yadatex	Mochila escolar de Dragon Ball.	17	Mochila yadatex dragon ball ...	assets/images/goku.jpg	Escolares
18	18	Profesional	535,20 €	moleskine	Libreta de raya edición Limita...	18	Libreta moleskine dragon ba...	assets/images/goku2.jpg	Escolares
19	19	Profesional	259,00 €	articolor	Bata para estudiantes. Son id...	19	Bata escolar articolor para l...	assets/images/bata.jpg	Escolares
20	20	Profesional	2.599,00 €	primus	Diadema ideal para jugar parti...	20	Diadema primus phs-250 ar...	assets/images/diadema.jpg	Oficina
21	21	Profesional	201,00 €	unico	Hojas con diversos colores y ...	21	Block 30x30cm unico con 3...	assets/images/hojas.jpg	Artes

Con esto ya podemos pasar a visualizar como funciona la función.

```
CREATE OR REPLACE FUNCTION public.buscar_utilidad(
    codigo1 integer)
RETURNS money
LANGUAGE 'sql'

COST 100
VOLATILE

AS $BODY$
SELECT precio
FROM producto
WHERE codigo=codigo1;
$BODY$;

ALTER FUNCTION public.buscar_utilidad(integer)
OWNER TO postgres;
```

Como se menciono anteriormente la función recibe un código y la función retorna un valor de tipo de dato money y con la ayuda del SELECT y WHERE podemos realizar la búsqueda de la utilidad donde el código sea igual al que se le pasa como parámetro a la función, de esa forma nos podrá dar



la utilidad de cualquier producto que se encuentre en la tabla producto únicamente con el código y se puede invocar en cualquier momento.

Query Editor

```
1  SELECT public.buscar_utilidad(2);
```

Data Output Query History Explain Messages Notifications

	buscar_utilidad	money
1	44,97 €	

- 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.

Para poner una solución a este punto importante del proyecto se implemento un procedimiento que recibiera el código y cantidad y como sabemos por la documentación los procedimientos son usualmente usados en las transacciones por su seguridad.

Para esto primero se analiza el orden de lo que se nos solicita realizar y con las estructuras de control se puede realizar fácilmente.

```
CREATE OR REPLACE PROCEDURE public.venta_articulo(
    cod integer,
    cantidad integer)
LANGUAGE 'plpgsql'
AS $BODY$  
DECLARE
    actual integer;
BEGIN
    UPDATE producto SET stock=stock-cantidad WHERE codigo=cod;
    SELECT stock INTO actual FROM producto WHERE codigo=cod;
    IF actual<1 THEN
        ROLLBACK;
        raise notice 'Hay menos de 3 en stock y no se realizó la transacción';
    ELSIF actual<3 THEN
        raise notice 'Hay menos de 3';
    ELSE
        COMMIT;
    END IF;
END;
$BODY$;
```



Podemos observar que se crea una variable de tipo entero para almacenar el valor del stock ya actualizado y esto dentro de la transacción, por lo cual ya podemos empezar con las estructuras de control y ir verificando cada uno, por ejemplo, si llega que actual que es la variable de guarda el valor de stock llega a ser menor a uno, se realiza un ROLLBACK y deshacemos la actualización que hicimos en el Stock y le mandamos un mensaje de que hay menos de 3 productos en Stock y por eso no se realiza la transacción y también se solicita que si hay menos de tres en stock se le haga saber, y con el ELSIF volvemos a revisar si es menor que tres y si lo es realiza la notificación, y al final guardamos los cambios con el COMMIT.

- Dada una fecha, o una fecha de inicio y fecha de fin, regresar la cantidad total que se vendió en esa fecha/periodo.

Para poner una solución a este punto importante del proyecto se implementó una función que recibiera la fecha y retornara la cantidad total que se vendió en ese fecha.

Para esto primero se analiza la tabla venta para identificar cuales son las columnas de fecha y precio total para poder saber que datos usar en la función.

Data Output						
	noventa [PK] integer	iva_apli integer	concepto character varying (100)	precio_total money	fecha_venta timestamp with time zone	rfc character varying (13)
1	2	16	[null]	20,00 €	2020-12-31 00:00:00-06	SUPL970118
2	3	16	[null]	10,00 €	2020-12-31 00:00:00-06	SUPL970118
3	4	16	[null]	30,00 €	2020-12-30 00:00:00-06	SUPL970118
4	5	16	hola	15,00 €	2021-01-04 00:00:00-06	SUPL970118
5	8	16	mama	15,00 €	2021-01-04 00:00:00-06	SUPL970118
6	10	16	mama	17,00 €	2021-01-04 00:00:00-06	SUPL970118
7	11	16	mama	10,00 €	2021-01-04 00:00:00-06	SUPL970118
8	25	16	mama	12,00 €	2021-01-04 00:00:00-06	SUPL970118
9	26	16	mama	12,00 €	2021-01-04 19:30:45.455652-06	SUPL970118
10	27	16	mama	12,00 €	2021-01-04 19:30:48.01783-06	SUPL970118
11	28	16	mama	12,00 €	2021-01-04 19:30:50.468204-06	SUPL970118
12	29	16	mama	12,00 €	2021-01-04 19:30:51.856112-06	SUPL970118
13	30	16	mama	12,00 €	2021-01-04 19:30:52.699353-06	SUPL970118
14	31	16	mama	12,00 €	2021-01-04 19:30:53.507196-06	SUPL970118
15	32	16	mama	12,00 €	2021-01-04 19:30:54.157753-06	SUPL970118
16	33	16	mama	12,00 €	2021-01-04 19:30:54.751363-06	SUPL970118
17	34	16	mama	12,00 €	2021-01-04 19:30:55.424867-06	SUPL970118

Ya con esto podemos pensar que necesitaremos usar funciones de agregación para que realice la suma de ciertos registros dada una fecha.



```
CREATE OR REPLACE FUNCTION public.cantidad_total_ventas(
    fecha_ingresada date)
RETURNS money
LANGUAGE 'sql'

COST 100
VOLATILE

AS $BODY$
SELECT SUM(precio_total) FROM venta WHERE fecha_venta=fecha_ingresada
$BODY$;

ALTER FUNCTION public.cantidad_total_ventas(date)
OWNER TO postgres;
```

Como podemos visualizar nuestra función va recibir un dato de tipo fecha y va regresar un dato de tipo money, para obtener la cantidad total de cierta fecha es necesario hacer uso del SELECT con una función de agregación y de un WHERE, para esto la función de agregación que vamos usar será "SUM" que nos permite sumar cada uno de los registros de la columna "precio_total" de la tabla venta donde la fecha

De esta forma obtenemos la suma de toda la cantidad que se obtuvo en cierta fecha, esta función es fácil de usar y acepta las fechas que se ingresen.

1	SELECT public.cantidad_total_ventas(
2	'31-12-2020'
3)

Data Output Query History Explain Messages Notifications

	cantidad_total_ventas	money	lock
1	30,00 €		

- Permitir obtener el nombre de aquellos productos de los cuales hay menos de 3 en stock.

En la solución de este punto se realizó una función que retornará una tabla con los nombres de los productos que tuvieran menos de tres en stock.

Para esto se necesita ver la tabla producto y ver las columnas que se van a ocupar en este caso las columnas a ocupar serán nombre y stock.



```
1 SELECT * FROM public.producto
2 ORDER BY codigo ASC
```

Data Output Query History Explain Messages Notifications

	codigo [PK] integer	modelo character varying (30)	precio money	marca character varying (30)	descripcion character varying (150)	stock integer	nombre character varying (50)	imagen character varying (100)	categoria character varying (100)
1		1 Básico	46,50 €	BIC	Pluma de tinta negra y punto ...	1	Paquete pluma y lápices Bic	assets/images/17875-515x5...	Escolares
2		2 Profesional	44,97 €	Scribe	Cuaderno de cuadro chico de ...	24	Cuaderno Profesional (cuadr...	assets/images/1-500x500.jpg	Escolares
3		3 Básico	46,50 €	BIC	Lápiz clásico turquesa del nú...	3	Lápiz Bic	assets/images/2-676x676.jpg	Escolares
4		4 Básico	17,00 €	Pelikan	Goma rectangular en un emp...	4	Goma de Migajón PELIKAN ...	assets/images/1-3-1046x104...	Escolares
5		5 Profesional	44,97 €	Scribe	Cuaderno de cuadro chico de ...	5	Cuaderno Profesional (cuadr...	assets/images/1-500x500.jpg	Escolares
6		6 Profesional	2.312,00 €	Koi	Estuche con 60 pastillas de a...	6	Acuarela	assets/images/3-500x500.jpg	Artes
7		7 Profesional	44,97 €	MOLOTOW	Cuaderno de 39X27 cm con 2...	7	Cuaderno de dibujo	assets/images/4-500x500.jpg	Artes
8		8 Profesional	46,50 €	Blanca nieves	Lápiz de color profesional de ...	8	Lápices de colores	assets/images/estuche-lapic...	Artes
9		9 Profesional	438,90 €	Pinto	Estuche de pinceles de nylon ...	9	Pinceles profesionales	assets/images/218590-500x...	Artes
10		10 Profesional	70,97 €	Ati	Godete desechable de plástic...	10	Godete	assets/images/436-500x500.j...	Artes
11		11 Profesional	572,18 €	Marc	Paquete de 60 rotuladores de ...	11	Rotuladores de colores mar...	assets/images/rotulador.jpg	Artes
12		12 Profesional	780,00 €	SPEEDBALL	Este conjunto de herramienta...	12	GUBIAS SPEEDBALL 4131 In...	assets/images/372.jpg	Artes
13		13 Profesional	5.190,00 €	REMBRANDT	Set de iniciación Rembrandt c...	13	Colores pastel REMBRANDT	assets/images/pasteles.jpg	Artes
14		14 Profesional	126,00 €	Moxy	Botella de 4.23 fl oz de esmal...	14	Pegamento con brillantina a...	assets/images/pega.jpg	Artes
15		15 Profesional	539,20 €	bostitch	Engrapadora de golpe para us...	15	Engrapadora de golpe bostit...	assets/images/engrapadora.j...	Oficina
16		16 Profesional	399,00 €	Art Deco	Libro de arte decó con los má...	16	Jumbo Slim Art Deco	assets/images/libro.jpg	Artes
17		17 Profesional	649,00 €	yadatex	Mochila escolar de Dragon Ball.	17	Mochila yadatex dragon ball ...	assets/images/goku.jpg	Escolares
18		18 Profesional	535,20 €	moleskine	Libreta de raya edición Limita...	18	Libreta moleskine dragon ba...	assets/images/goku2.jpg	Escolares
19		19 Profesional	259,00 €	articolor	Bata para estudiantes. Son id...	19	Bata escolar articolor para l...	assets/images/bata.jpg	Escolares
20		20 Profesional	2.599,00 €	primus	Diadema ideal para jugar parti...	20	Diadema primus phs-250 ar...	assets/images/diadema.jpg	Oficina
21		21 Profesional	201,00 €	unico	Hojas con diversos colores y ...	21	Block 30x30cm unico con 3...	assets/images/hojas.jpg	Artes

Ya con esto podremos pasar a la creación de la función que no tendrá parametros y retornara una tabla con solo la columna nombre, y con ayuda de un SELECT y un WHERE obtendremos lo que necesitamos, solo le decimos que seleccione el nombre de la tabla producto donde el stock sea menor a 3, y con eso nos regresará la tabla con los datos que queremos.

```
CREATE OR REPLACE FUNCTION public.buscar_nombre
)
RETURNS TABLE(nombre character varying)
LANGUAGE 'sql'

COST 100
VOLATILE
ROWS 1000

AS $BODY$
SELECT nombre
FROM producto
WHERE stock<3;
$BODY$;

ALTER FUNCTION public.buscar_nombre()
OWNER TO postgres;
```



Su utilización es muy sencilla, solo se llama a la función y retorna una tabla con los nombres de los productos que cumplen de condición de tener menos de tres en stock.

Query Editor

```
1  SELECT public.buscar_nombre()
```

Data Output Query History Explain Messages

	buscar_nombre
1	Paquete pluma y lápices Bic

- De manera automática se genere una vista que contenga información necesaria para asemejarse a una factura de una compra.

En este punto solamente pensamos que los datos para generar una factura ya los tenemos en diferentes tablas, solamente tomamos dos valores que son el RFC y el numero de venta (le llamamos noventa) y ya con esos datos hicimos un SELECT con los datos del cliente y de su compra, en este caso las tablas son cliente y venta respectivamente, y con ayuda de un WHERE solo hicimos que el número de venta coincidiera con el número de venta ingresado como parámetro en la función y lo mismo con el RFC que coincidiera, ya con eso podemos realizar una selección de datos para el RFC y esta función nos regresa la información en orden de tabla, llamando al objeto.

```
CREATE OR REPLACE FUNCTION public.genera_factura_(
    rfc1 character varying,
    noventa integer)
RETURNS TABLE(nombre character varying, rfc character varying, colonia character varying, calle character varying, num_interior character varying, num_exterior
LANGUAGE 'sql'

COST 100
VOLATILE
ROWS 1000

AS $BODY$
SELECT c.nombre, c.rfc, c.colonia, c.calle, c.num_interior,c.num_exterior, c.cp, v.noventa, v.precio_total, v.fecha_venta
From cliente c, venta v
where c.rfc=rfc1 and v.noventa=noVenta;
$BODY$;
```

- Crear al menos, un índice, del tipo que se prefiera y donde se prefiera. Justificar el porqué de la elección en ambos aspectos.



6.2. Creación del índice

Durante la creación de la tabla cliente se creó un índice a la tabla cliente, este índice llamado 'nombre' es de tipo b-tree. Este índice se creó en cliente con el objetivo de facilitar las operaciones sobre la tabla, así como eficientiza las consultas por medio de la búsqueda de nombre.

La captura de pantalla muestra la interfaz de pgAdmin III. En el panel izquierdo, se observa una jerarquía de objetos: Tables (12), cliente (que contiene Columns (8) incluyendo rfc, razon_soc, cp, colonia, calle, nombre, num_exterior y num_interior), Constraints, Indexes (3) (destacada en azul) y RLS Policies. Los tres índices creados son:

```
30
31 CREATE INDEX "Nombre"
32   ON public.cliente USING btree
33   (nombre COLLATE pg_catalog."default" ASC NULLS LAST)
34   TABLESPACE pg_default;
35 -- Index: fkj_cliente_cp_FK
36
37 -- DROP INDEX public."fkj_cliente_cp_FK";
38
39 CREATE INDEX "fkj_cliente_cp_FK"
40   ON public.cliente USING btree;
```

En la parte inferior, se muestra el resultado de la ejecución: "CREATE INDEX" y "Query returned successfully in 44 msec."

De esta manera la información del índice puede hacer un recorrido de árbol por los datos hasta que encuentra un nodo hoja con alguna coincidencia. Este tipo de recorrido es una operación eficiente, tanto que las claves primarias en son un índice de tipo b-tree desde su creación.

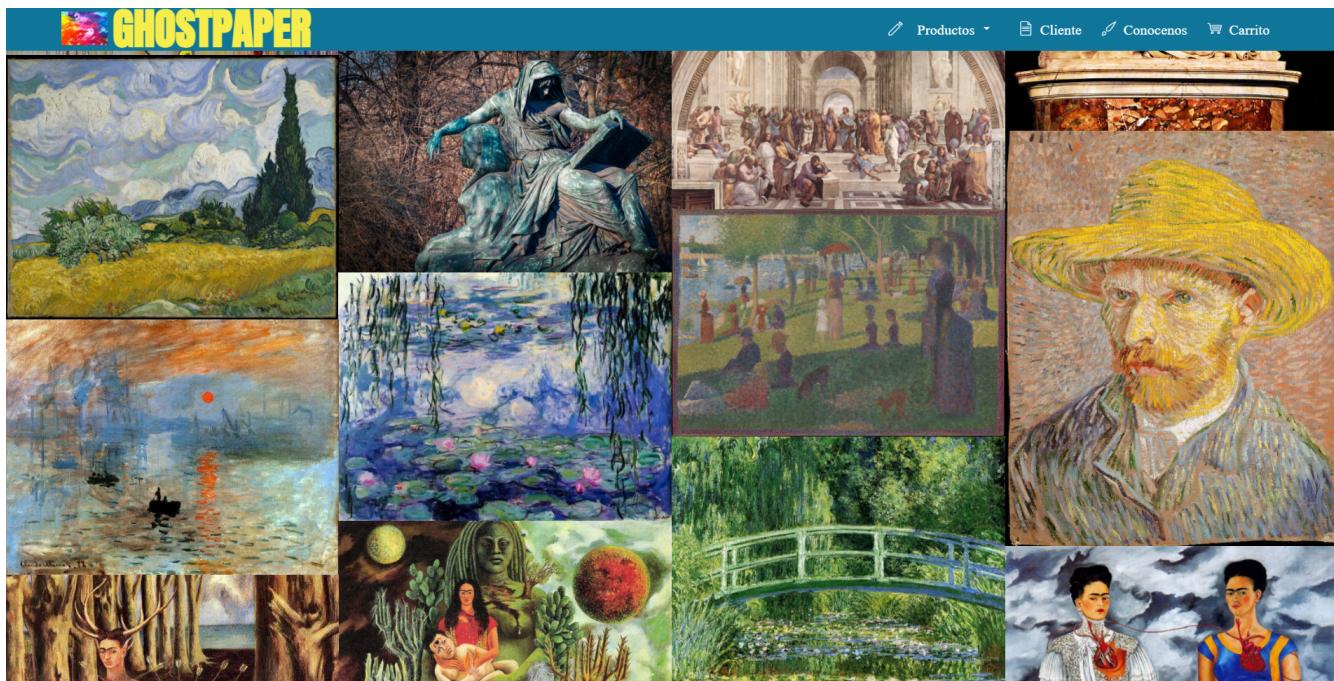
6.3. Implementación de la Base de datos a una página Web

Para el desarrollo de una interfaz gráfica se optó por una página web, puesto que su implementación es más sencilla y rápida en su diseño. Para el diseño se utilizó una aplicación freeware de diseño web llamada "Mobirise". Se optó por usar esto ya que la prioridad del proyecto era enfocarnos en la conexión de la base de datos y las funcionalidades que debe de tener.

La página web tiene un diseño moderno con un fondo abstracto de colores rojos, amarillos y azules. En la parte superior, hay un menú horizontal con íconos para Productos, Cliente, Conocenos y Carrito. El logo "GHOSTPAPER" aparece en la parte superior izquierda. El centro de la página tiene un título grande "Bienvenido a la Papelería Ghost Paper" y un párrafo descriptivo: "Somos una papelería inspirada en tí para darte las herramientas necesarias para estimular tu creatividad y tu bolsillo. Asegurando productos ecológicos y reciclados para la conservación de nuestro mundo." Abajo del título, hay tres botones redondos: "Sobre nosotros" (rojo), "Ofertas" (amarillo) y "Manualidades" (verde).



Tomando el tema de una papelería con un enfoque de creación artística se implementó una sección en la que el visitante puede encontrar inspiración artística en una ciber-galería, así como también puede aprender a pintar con videos de Bob Ross.





Posteriormente se programó el carrito de compra con Javascript, tomando como base código fuente que generó mobirise.

La captura de pantalla muestra la página principal de Ghostpaper. En la parte superior, hay un menú con enlaces a "Productos", "Cliente", "Conocenos" y "Carrito". El fondo es una imagen abstracta de colores amarillo, verde y azul. Se presentan seis productos en tarjetas:

- Godete**: Godete desecharable de plástico circular con tapa transparente, adecuado para preparar, diluir y mezclar colores. Precio: 70.97 \$. Botón: "Añadir al carrito".
- Rotuladores de colores marca Marc**: Paquete de 60 rotuladores de colores de excelente calidad. Precio: 572.18 \$. Botón: "Añadir al carrito".
- GUBIAS SPEEDBALL 4131 Intercambiables**: Este conjunto de herramientas de linóleo contiene un mango de lino y cortadores de linóleo 1, 2, 3, 5, 6. Precio: 780 \$. Botón: "Añadir al carrito".
- Colores pastel REMBRANDT**: Imagen de una paleta de colores pastel.
- Pegamento con brillantina american crafts**: Imagen de una botella de pegamento.
- Engrapadora de golpe bostitch**: Imagen de una engrapadora.

La captura de pantalla muestra la página del carrito de compra de Ghostpaper. En la parte superior, hay un menú con enlaces a "Productos", "Cliente", "Conocenos" y "Carrito". El fondo es una imagen de un organizador de maquillaje con secciones coloridas. Un cuadro central titulado "Carrito" muestra los siguientes ítems:

Ítem	Precio	Opción X
1 x Mochila yadatex dragon ball super Goku	649 \$	X
1 x Jumbo Slim Art Deco	399 \$	X
1 x Diádem primus phs-250 arcus 7.1 ch	2599 \$	X
1 x Block 30x30cm unico con 33 hojas	201 \$	X

Total: 3849.00€

Botones: "Vaciar" y un icono de flecha.



6.4. Programación en Javascript

Para manejar un conjunto de datos y operarlos, se programó por medio de Javascript el carrito. Se tomó como base el html y se declaró un apartado de javascript en el que se programó un código el cual se carga junto con la página.

Código del carrito escrito en javascript:

```
<script>
window.onload = function () {
    // Variables

    let baseDeDatos = [
        {
            id: 1,
            nombre: 'Paquete pluma y lápices Bic',
            precio: 46.50,
            descripcion:'Pluma de tinta negra y punto mediano. En un empaque con 4 lápices del número 2.',
            imagen: 'assets/images/17875-515x515.png'
        },
        {
            id: 2,
            nombre: 'Cuaderno Profesional (cuadro chico)',
            precio: 44.97,
            descripcion:'Cuaderno de cuadro chico de 90 Hojas.',
            imagen: 'assets/images/1-500x500.jpg'
        },
        {
            id: 3,
            nombre: 'Lápiz Bic',
            precio: 46.50,
            descripcion:'Lápiz clásico turquesa del número 2. En un empaque de 8 piezas(Incluye sacapuntas básico).',
            imagen: 'assets/images/2-676x676.jpg'
        },
        {
            id: 4,
            nombre: 'Goma de Migajón PELIKAN STRIKE ',
            precio: 17,
            descripcion:'-Goma rectangular en un empaque de 2 Piezas.',
            imagen: 'assets/images/1-3-1046x1046.png'
        },
        {
            id: 5,
            nombre: 'Cuaderno Profesional (cuadro grande) ',
            precio: 44.97,
            descripcion:'Cuaderno de cuadro chico de 90 Hojas.',
            imagen: 'assets/images/1-500x500.jpg'
        },
        {
            id: 6,
            nombre: 'Acuarela',
            precio: 2312.00,
            descripcion:'Estuche con 60 pastillas de acuarela profesional con colores vibrantes.',
            imagen: 'assets/images/3-500x500.jpg'
        },
        {
            id: 7,
            nombre: 'Cuaderno de dibujo',
            precio: 44.97,
        }
    ]
}
```

Primero, en el archivo html de la página de carrito, se declara en un apartado de script un ArrayList con los rubros necesarios para presentar los productos necesarios. Estos rubros son id,nombre, precio, descripción y dirección de imagen (el cuál señalará al html dónde extraer la imagen).

```
let $escolar = document.querySelector('#escolar');
let $dibujo = document.querySelector('#dibujo');
let carrito = [];
let total = 0;
let $carrito = document.querySelector('#carrito');
let $total = document.querySelector('#total');
let $botonVaciar = document.querySelector('#boton-vaciar');
let $compra = document.querySelector('#compra');
```



Luego se extrajeron e inicializaron los valores de las variables relacionadas con el carrito, a las cuales posteriormente se les asignarán los valores pertinentes dependiendo de las funciones que se ejecutarán durante la carga de la página.

```
// Funciones
function renderItems() {
    for (let info of baseDeDatos) {
        // Estructura
        let miNodo = document.createElement('div');
        miNodo.classList.add('card', 'col-sm-4');
        // Body
        let miNodoCardBody = document.createElement('div');
        miNodoCardBody.classList.add('card-body');
        // Titulo
        let miNodoTitle = document.createElement('h5');
        miNodoTitle.classList.add('card-title');
        miNodoTitle.textContent = info['nombre'];
        // Imagen
        let miNodoImagen = document.createElement('img');
        miNodoImagen.classList.add('img-fluid');
        miNodoImagen.setAttribute('src', info['imagen']);
        // Precio
        let miNodoPrecio = document.createElement('p');
        miNodoPrecio.classList.add('card-text');
        miNodoPrecio.textContent = info['precio'] + ' $';
        //descripcion
        let miNodoDescripcion = document.createElement('p');
        miNodoDescripcion.classList.add('card-text');
        miNodoDescripcion.textContent = info['descripcion'];
        // Boton
        let miNodoBoton = document.createElement('button');
        miNodoBoton.classList.add('btn', 'btn-primary', 'display-4');
        miNodoBoton.textContent = 'Añadir a carrito';
        miNodoBoton.setAttribute('marcador', info['id']);
        miNodoBoton.addEventListener('click', anyadirCarrito);
        // Insertamos
        miNodoCardBody.appendChild(miNodoImagen);
        miNodoCardBody.appendChild(miNodoTitle);
        miNodoCardBody.appendChild(miNodoDescripcion);
        miNodoCardBody.appendChild(miNodoPrecio);
        miNodoCardBody.appendChild(miNodoBoton);
        miNodo.appendChild(miNodoCardBody);
        miNodoCardBody.classList.add('card-wrapper');
        $escolar.appendChild(miNodo);
    }
}
```

Aquí se genera el botón de compra de cada producto así como su información y su imagen en html. Todo esto en base a la lista de productos anteriormente mencionada.

```
function renderizarCarrito() {
    // Vaciamos todo el html
    $carrito.textContent = '';
    // Quitamos los duplicados
    let carritoSinDuplicados = [...new Set(carrito)];
    // Generamos los Nodos a partir de carrito
    carritoSinDuplicados.forEach(function (item, indice) {
        // Obtenemos el item que necesitamos de la variable base de datos
        let miItem = baseDeDatos.filter(function(itemBaseDatos) {
            return itemBaseDatos['id'] == item;
        });
        // Cuenta el número de veces que se repite el producto
        let numeroUnidadesItem = carrito.reduce(function (total, itemId) {
            return itemId === item ? total += 1 : total;
        }, 0);
        // Creamos el nodo del item del carrito
        let miNodo = document.createElement('li');
        miNodo.classList.add('list-group-item', 'text-right', 'mx-2');
        miNodo.textContent = `${numeroUnidadesItem} x ${miItem[0]['nombre']} - ${miItem[0]['precio']} $`;
        // Boton de borrar
        let miBoton = document.createElement('button');
        miBoton.classList.add('btn', 'btn-danger', 'mx-5');
        miBoton.textContent = 'X';
        miBoton.style.marginLeft = '1rem';
        miBoton.setAttribute('item', item);
        miBoton.addEventListener('click', borrarItemCarrito);
        // Mezclamos nodos
        miNodo.appendChild(miBoton);
        $carrito.appendChild(miNodo);
    });
}
```



Aquí se genera el carrito de compras. Aquí muestra el contenido de la lista y le da formato a la salida en html.

Finalmente, estas son las funciones que manejan el evento onclick de los botones creados en los segmentos de código anteriores:

```
function anyadirCarrito () {
    // Anyadimos el Nodo a nuestro carrito
    carrito.push(this.getAttribute('marcador'))
    // Calculo el total
    calcularTotal();
    // Renderizamos el carrito
    renderizarCarrito();
}
```

```
function borrarItemCarrito() {
    // Obtenemos el producto ID que hay en el botón pulsado
    let id = this.getAttribute('item');
    // Borramos todos los productos
    carrito = carrito.filter(function (carritoId) {
        return carritoId !== id;
    });
    // volvemos a renderizar
    renderizarCarrito();
    // Calculamos de nuevo el precio
    calcularTotal();
}

function calcularTotal() {
    // Limpiamos precio anterior
    total = 0;
    // Recorremos el array del carrito
    for (let item of carrito) {
        // De cada elemento obtenemos su precio
        let miItem = baseDeDatos.filter(function(itemBaseDatos) {
            return itemBaseDatos['id'] == item;
        });
        total = total + miItem[0]['precio'];
    }
    // Formateamos el total para que solo tenga dos decimales
    let totalDosDecimales = total.toFixed(2);
    // Renderizamos el precio en el HTML
    $total.textContent = totalDosDecimales;
}

function vaciarCarrito() {
    // Limpiamos los productos guardados
    carrito = [];
    // Renderizamos los cambios
    renderizarCarrito();
    calcularTotal();
}
```



Se manejó otra función necesaria para el evento de compra pero como este tiene que ver con la conexión de la base de datos, se retomará más adelante.

6.5. Conexión de base de datos a partir de la página web

Finalmente se decidió conectar a la base de datos, se empezó evaluando la correcta conexión a la base por medio de consultas en el banner de inicio. Para esto se hicieron pequeñas consultas como:

```
<?php
$conexion=pg_connect("host=localhost port=5432 dbname=GHOSTPAPER user=postgres password=987654321");
if(!$conexion){
echo "Error";
}else{
$sql = pg_query($conexion,"SELECT nombre from Proveedores");
if(!$sql){
echo "Proximamente nuevos proveedores";
exit;
}else{
echo "Nuestros proveedores: <br><br>";
while( $obj = pg_fetch_row($sql)){
echo $obj[0];
echo "<br>";
}
}
}?
}?
```



GHSTPAPER

Productos Cliente Conocenos Carrito

Nuestros 3 productos con más ventas

*Jumbo Slim Art Deco

*Cuaderno de dibujo

*Engrapadora de golpe bostitch



```
<?php
$conexion=pg_connect("host=localhost port=5432 dbname=GHOSTPAPER user=postgres password=987654321");
if(!$conexion){
echo "Error";
}else{
$sql = pg_query($conexion,"SELECT P.nombre AS VENTA
                           FROM Producto P
                           INNER JOIN CONSTA C
                           ON P.CODIGO=C.CODIGO_FK
                           GROUP BY P.nombre
                           Limit 3");
if(!$sql){
echo "Proximamente nuevos productos";
exit;
}else{
echo "Nuestros 3 productos con más ventas <br><br>";
while( $obj = pg_fetch_row($sql)){
    echo "*";
    echo $obj[0] ;
    echo "<br>";
}}
}?
```



Cabe destacar que para conectar la página desde html se utilizó php para conectarse a la base de datos por medio del usuario postgres.

Lo siguiente a conectar fue el formulario de registro de cliente, para ese sólo se tomaban los valores registrados en el html y con el método POST se tomaban los valores y se enviaba a la consulta si estos datos poseían el formato requerido.

```
<div class="container">
    <div class="row justify-content-center">
        <!--Formbuilder Form-->
        <form action="conexion.php" method="POST" class="mbr-form form-with-styler"><input type="hidden" name="email" data-form-email="true">
            <div class="row row-sm-offset">
                <div hidden="" data-form-alert="" class="alert alert-success col-12">Información registrada</div>
                <div hidden="" data-form-alert-danger="" class="alert alert-danger col-12"></div>
            </div>
            <div class="dragArea row row-sm-offset">
                <div class="col-md-4 form-group" data-for="name">
                    <label for="name-form1-v" class="form-control-label mbr-fonts-style display-7">Nombre</label>
                    <input type="text" name="Nombre" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group" data-for="email">
                    <label for="email-form1-v" class="form-control-label mbr-fonts-style display-7">RFC</label>
                    <input type="text" name="rfc" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group">
                    <label for="phone-form1-v" class="form-control-label mbr-fonts-style display-7">Teléfono</label>
                    <input type="tel" name="telefono" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group">
                    <label for="phone-form1-v" class="form-control-label mbr-fonts-style display-7">Correo</label>
                    <input type="email" name="correo" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group">
                    <label for="phone-form1-v" class="form-control-label mbr-fonts-style display-7">CP</label>
                    <input type="number" name="cp" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group">
                    <label for="phone-form1-v" class="form-control-label mbr-fonts-style display-7">Colonia</label>
                    <input type="text" name="colonia" class="form-control display-7" >
                </div>
                <div class="col-md-4 form-group">
                    <label for="phone-form1-v" class="form-control-label mbr-fonts-style display-7">Calle</label>
                    <input type="text" name="calle" class="form-control display-7" >
                </div>
            </div>
        </form>
    </div>
</div>
```

Para el carrito no se utilizó el método POST, pues se tenía que recoger los valores del carrito el cual está escrito en java. Para conectar de Java a php (que es dónde se hacen las consultas) se tuvo que usar la función de getElementById() esta función obtiene los valores insertados en html y se abre una nueva ventana a la página de html con el php, la cual posee en sus argumentos los valores obtenidos del html junto con el valor del carrito como valores concatenados a la dirección de la página de la conexión.



Tu compra ha sido realizada ¡Muchas gracias por tu preferencia!
Vuelve pronto sandra susana

Información de tu compra:

Venta:
Número de venta ---> VENTA-00331
IVA aplicado --->16%
concepto --->''
precio total --->3397.17
fecha --->24.01.2021
dirección de envío --->juanchita

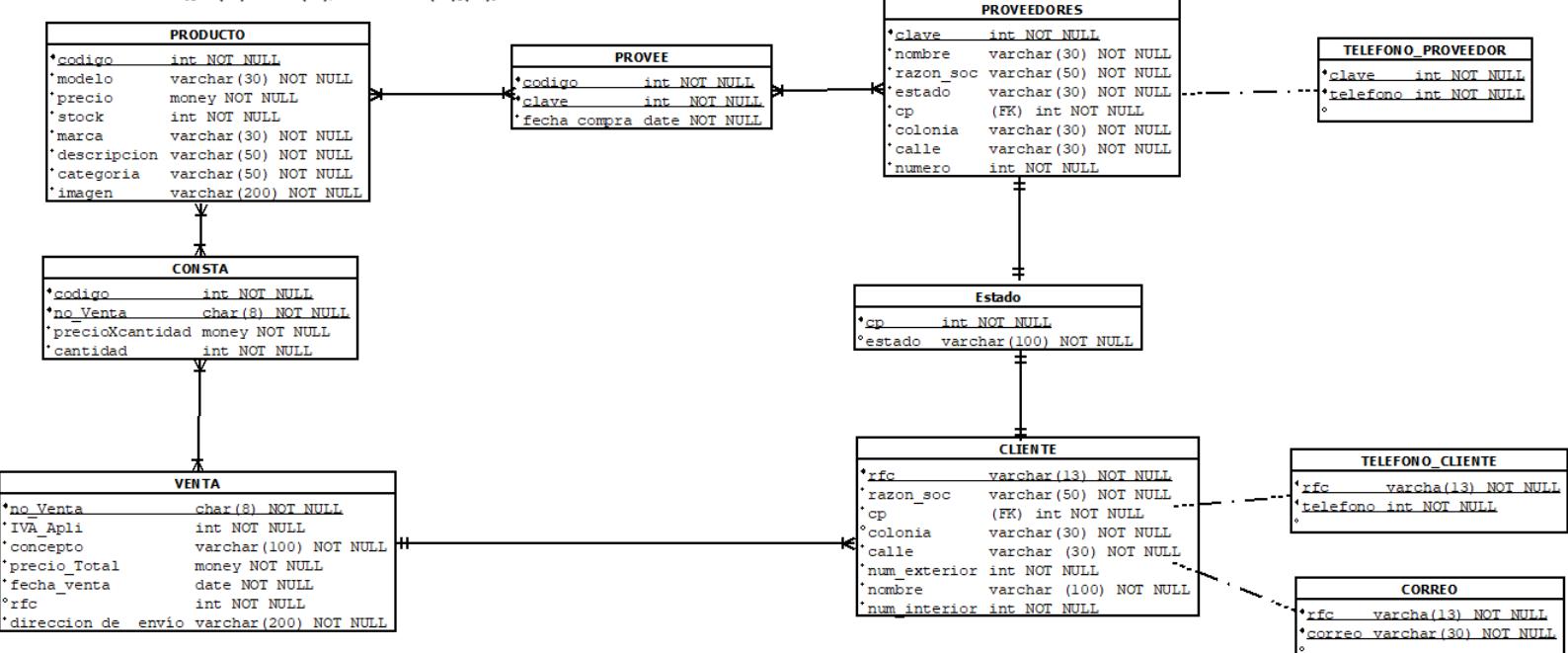
Factura:
Nombre completo --->SANDRA
RFC del cliente --->PEG59811278
Razon social ---> null '
dirección del cliente ---> CP: 6700,colonia: ROMA,calle: CHIHUHUA,
número exterior: 123, número interior: 2C, ciudad: Ciudad de México
Número de venta ---> VENTA-00331'
precio total --->3397.17
fecha --->24.01.2021
dirección de envío --->juanchita

6.6. Modificaciones finales de la base de datos

Por razones de implementación se decidió añadir atributos extra.

Modelo Relacional

```
PRODUCTO:{codigo(PK) int,modelo varchar(30),precio money,stock int,marca varchar(30),descripcion varchar(50)}  
PROVEEDOR:{dave (PK) int,nombre varchar(100),razon_soc varchar(50),estado varchar(30),cp int,colonia varchar(30),calle varchar(30),numero int}  
VENTA:{no_Venta (PK) char(8),IVA_Apli int,concepto varchar(100),precio_Total money,fecha_venta date}  
CLIENTE:{rfc (PK) varchar(13),razon_soc varchar(50),estado varchar(30),cp int,colonia varchar(30),calle varchar(30),numero int,nombre varchar(100)}  
PROVEE:{[codigo (FK) int, clave (Fk) int](PK),fecha_compra date}  
CONSTA:{[codigo (FK) int, no_venta (FK) char(8)](PK),precio money,cantidad int}  
TELEFONO_PROVEEDOR:{[clave(FK) int, telefono int](PK)}  
TELEFONO_CLIENTE:{[rfc(FK) varchar(13), telefono int](PK)}  
EMAIL:{[rfc (FK) varchar(13), email varchar(30)](PK)}
```





7. Presentación

7.1. Conexión de la base de datos a la página web

En esta parte del proyecto se toma las comunicaciones entre la base de datos y la página web, y para lograrlo es necesario contar con un servidor web que se encargue de dichas comunicaciones, explicando brevemente el software que usamos es XAMPP, que tiene un conjunto de modulos cargados que permite la comunicación de diversas manejadores de bases de datos, y lo escogimos por ser de licencia libre por uso no comercial por el momento.

Entre los diversos modulos que tiene esta el "php.ini.en donde tenemos que habilitar la extensión de postgres para apache, pueda usas los modulos de php con postgres, por defecto este aparece deshabilitado pero simplemente debemos quitar los ";.en las dos extenciones .extension=pdo_pgsql" y "extension =pgsql" y arealizado es solo se guarda el archivo, de la siguiente forma :

```
php.ini: Bloc de notas
Archivo Edición Formato Ve
;
; Notes for Windows env
;
; - Many DLL files are
;   extension folders a
;   Be sure to appropri
;
extension=bz2
extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
;extension=gd
extension=gettext
;extension=gmp
;extension=intl
;extension=imap
;extension=ldap
extension=mbstring
extension=exif      ; M
extension=mysqli
;extension=oci8_12c  ;
;extension=odbc
;extension=openssl
;extension=pdo_firebird
extension=pdo_mysql
;extension=pdo_oci
;extension=pdo_odbc
extension=pdo_pgsql
extension=pdo_sqlite
extension=pgsql
;extension=shmop
```

Y en el archivo "httpd.conf" se agregan los tipos de datos que se van a estar utilizando en caso de ser necesario, es este caso en particular fue necesario porque combinamos javascript con html, es



por eso que en nuestro caso particular realizamos estas acciones:

httpd.conf: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
AddType application/x-httpd-php .html .php
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
```

Con estas modificaciones no tendremos problemas de comunicación entre apache y la página web, ahora es necesario configurar las entradas de conexión entre la base de datos y la página por medio del Apache usando funciones de php.

```
$conexion=pg_connect("host=localhost port=5432 dbname=GHOSTPAPER user=postgres password=987654321");
```

Con esa función "pgconectrealizamos la conexión a la base de datos, y podemos observar que los campos que nos muestra son similares a los datos que nos pide el manejador cuando ingresamos desde consola, en esta función debemos especificar lo siguiente en este orden:

1. host: Ya sea por ip o en este caso que es una ip local, podemos usar 'localhost' que es donde se conecta la base de datos.
2. port: Especificar el puerto 5432 que es el que usa postgres o en caso de usar algún otro se especifica.
3. dbname: Ingresar el nombre de la base de datos a la cual nos vamos a conectar.
4. user: El nombre con el usuario que se va conectar
5. password: La contraseña de dicho usuario.

Ya con esto podremos conectar la base de datos con la página web.



7.2. Cliente servidor

Para esta parte se necesita saber la ip que se va utilizar para la conexión, en este caso por la estructura que nuestro servidor de internet se hará la conexión cliente servidor con ayuda de una maquina virtual, la cual va funcionar como cliente, y la maquina principal funcionara como servidor.

Para saber nuestra ip en el caso de "Windows 10." es necesario ingresar al CMD y ingresar el comando ipconfig, el cual nos dará información de configuración de nuestras ip, la publica y la local, con "Bunsen Lab'scon el comando ip addr", obtendremos en particular la ip publica y la colocaremos en el archivo pg_hba.conf de la máquina principal junto con las máscaras y el tipo de acceso.

De esa forma le estaremos diciendo al manejador que permita las conexiones desde esa ip.

```
pg_hba.conf: Bloc de notas
Archivo Edición Formato Ver Ayuda

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all scram-sha-256
host all all 127.0.0.1/8 md5
host all all 192.168.0.16/24 md5
host all all 0.0.0.0/0 scram-sha-256
# IPv4 local connections:
host all all 127.0.0.1/32 scram-sha-256
# IPv6 local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all scram-sha-256
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
```

Y para ver los detalles del ip en linux:

```
serveradmin@bl64:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ff:43:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.16/24 brd 192.168.0.255 scope global dynamic enp0s3
        valid_lft 3018sec preferred_lft 3018sec
    inet6 fe80::a00:27ff:fe43:de/64 scope link
        valid_lft forever preferred_lft forever
```



De esa forma debe quedar configurado, después solo se guarda y se reinicia el servicio de postgres para no tener complicaciones, en algunos casos no es necesario.

Es importante saber a que ip nos vamos a conectar desde la maquina virtual, por se cliente es necesario escribir el host, en donde va la ip de la maquina principal, de la siguiente manera:

```
serveradmin@bl64:/usr/lib/postgresql/12/bin$ ./psql -U postgres -W -h 192.168.0.9 -p 5432
Contrasena:
psql (12.2 (Debian 12.2-1.pgdg90+1), servidor 13.0)
ADVERTENCIA: psql versión mayor 12, servidor versión mayor 13.
          Algunas características de psql podrían no funcionar.
Digite «help» para obtener ayuda.

postgres=#
```



Especificando el usuario, la contraseña, el host al que se va conectar, es decir, la ip del servidor y su respectivo puerto.

Podemos verificar que se realizo la conexión verificando las bases que se tienen, deben ser las que tiene el servidor, en este caso la maquina principal.

Listado de base de datos						
Nombre	Dueño	Codificación	Collate	Ctype	Superuser	Main Menu
egios					+Tab	Client M
GHOSTPAPER	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252		Editor
datos_clase	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252		Media Player
postgres	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252		Web Browser
prueba	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252		Task Manager
registro_vuelos	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252		
template0	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252	=c/postgres	
+						
c/postgres						
template1	postgres	UTF8	Spanish_Spain.1252	Spanish_Spain.1252	=c/postgres	
+						
c/postgres						
(7 filas)						

Vemos que es así y ahora procedemos a realizar la conexión a GHOSTPAPER, de la siguiente manera:

```
postgres=# \c GHOSTPAPER;
Contraseña para usuario postgres:
psql (12.2 (Debian 12.2-1.pgdg90+1), servidor 13.0)
ADVERTENCIA: psql versión mayor 12, servidor versión mayor 13.
      Algunas características de psql podrían no funcionar.
Ahora está conectado a la base de datos «GHOSTPAPER» con el usuario «postgres».
GHOSTPAPER=#
```

Ahora realizamos una pequeña consulta para verificar que este mostrando los registros esperados.



1 Básico	46,50 € BIC		Pluma de tinta negra y punto mediano. En un e	
mpaque con 4 lápices del número 2.			assets/images/17875-515x515.png	
1 Paquete pluma y lápices Bic			Cuaderno de cuadro chico de 90 Hojas.	
Escolares			assets/images/1-500x500.jpg	
2 Profesional	44,97 € Scribe		Lápiz clásico turquesa del número 2. En un em	
			assets/images/2-676x676.jpg	
24 Cuaderno Profesional (cuadro chico)			-Goma rectangular en un empaque de 2 Piezas.	
Escolares			assets/images/1-3-1046x1046.png	
3 Básico	46,50 € BIC		Cuaderno de cuadro chico de 90 Hojas.	
paque de 8 piezas(Incluye sacapuntas básico).			assets/images/1-500x500.jpg	
3 Lápiz Bic			Estuche con 60 pastillas de acuarela profesio	
Escolares			assets/images/3-500x500.jpg	
4 Básico	17,00 € Pelikan		Cuaderno de 39X27 cm con 24 hojas.	
			assets/images/4-500x500.jpg	
4 Goma de Migajón PELIKAN STRIKE				
Escolares				
5 Profesional	44,97 € Scribe			
5 Cuaderno Profesional (cuadro grande)				
Escolares				
6 Profesional	2.312,00 € Koi			
onal con colores vibrantes.				
6 Acuarela				
Artes				
7 Profesional	44,97 € MOLOTOW			
7 Cuaderno de dibujo				

En efecto la conexión cliente servidor fue un éxito, podemos apreciar que la comunicación se realiza satisfactoriamente.



8. Conclusiones

- Pérez Gutiérrez Sandra Susana.

Considero que se cumplieron los objetivos del proyecto. Se desarrolló e implementó una base de datos a partir de los requerimientos planteados en una situación la cual puede llegar a ser una situación real y cotidiana en nuestra rama de ingeniería en computación.

Se aprendió sobre la práctica de conceptos de ingeniería de software, bases de datos, estructuras de datos y algoritmos. Durante el desarrollo de la etapa de diseño se aplicaron las etapas del modelo entidad relación, el modelo relacional y la normalización (en esta etapa se filtraron redundancias y se aplicaron los conceptos vistos en clase), luego se implementó la base de datos (en este punto se declararon índices y funciones, así como se aplicó lo visto en la parte práctica de la clase y en laboratorio) después se hizo el diseño y conexión de la página web, durante esta etapa se programó el carrito y se conectó a la base de datos usando php.

En la conexión a la base de datos se debió de modificar algunas configuraciones especiales para el manejador de la base de datos que usamos (en este caso postgresql) en las configuraciones de apache del servidor web xampp, el cual también actúa como interprete de php.

Se utilizó html para el front-end de la página, para la funcionalidad se usó javascript, con el cuál en conjunto con html generan una aplicación web; para la conexión el php es fundamental para manejar cualquier base de datos pues posee muchísimas funciones especializadas en esto.

Se pudo observar con este trabajo la utilidad y complejidad de una base de datos, que aunque nuestra base de datos fue sencilla sirve como ejemplo práctico de lo que conlleva una base de datos y su implementación web. Así mismo se concluye que una base de datos es muy útil para almacenar datos de manera estructurada y ordenada. Con ella se puede obtener, a partir de datos, información estructurada y ordenada que permite tomar decisiones y evaluar resultados. Esta función es vital para la mantenibilidad y toma de decisiones económicas de un negocio, para el funcionamiento de una aplicación, resguardo de la información de clientes,etc.

- Suxo Pérez Luis Axel

Se cumplieron con los objetivos del proyecto y se utilizaron la mayoría de los temas vistos en clase y en el laboratorio, y se puede decir que este proyecto es un ejemplo de lo que se pide en el área laboral, y también nos puede llegar a servir en un futuro cercano.

En este proyecto se utilizaron demasiado las funciones para la solución de la mayoría de sus puntos y un procedimiento para la transacción, esto es de gran importancia dado que usarlas nos da una buena herramienta ya sea que la función sea básica o compleja, incluso si sabemos que es un query de uso constante lo podemos pasar a una función para ahorrarnos líneas de código y solamente invocarla, y también las funciones de agregación se usaron y son de gran utilidad.

Los procedimientos tienen una gran importancia en el momento de hacer transacciones, porque estos mantienen más seguras las transacciones, y como lo vimos en el proyecto se nos permite el uso



de las estructuras de control en funciones como en los procedimientos, de hecho los procedimientos en postgres en algo nuevo porque se habilito desde la versión 9, y en lo personal siento que postgres tiene mucho potencial y probablemente en cierto punto pueda ser comprado por alguna empresa.

Y volviendo a lo mas básico el diseño de una base de datos es de mucha importancia para muchas cosas, esto define si la base de datos va a ser utilidad y como diseñadores debemos tener presente lo que se desea hacer y si se es posible mejorarla para que sea rentable para empresas u organizaciones.

Por último y no menos importante conocer el entorno de trabajo, tablas, índices, procedimientos, funciones, etc. es de gran importancia antes de modificar o realizar algún movimiento en alguna base de datos, como también conocer los archivos pg_hba.conf y postgresql.conf para conocer los métodos de autenticación y si se requiere configurar más conexiones, y para saber la configuración de conexiones, ubicaciones, configuraciones en si del manejador respectivamente, y lo que siempre deberíamos hacer un respaldo de la base de datos por si se presenta algún problema, volver a habilitar la base de datos y no perder los datos.



9. Referencias

1. Microsoft, "Restricciones entre claves principales y claves externas", <https://cutt.ly/ChmFTCN>, 25 de enero de 2021.
2. tutorialesdeprogramacionya, "Restricciones foreign key (acciones)", <https://cutt.ly/chmFOGT>, 25 de enero de 2021.
3. tutorialesdeprogramacionya, "Funciones SQL", <https://cutt.ly/mjZgQJc>, 25 de enero de 2021.
4. postgresql, "Create procedure", <https://cutt.ly/FjZQU4W>, 25 de enero de 2021.
5. postgresql, "Control structures", <https://cutt.ly/QjZRUhy>, 25 de enero de 2021.
6. apachefriends, "Descargar", <https://cutt.ly/qjMiSUq>, 25 de enero de 2021.
7. php, "PostgreSQL", <https://cutt.ly/GjMiAmg>, 25 de enero de 2021.