Proyecto Bases de Datos

Mejia Ramos Bryan
García Sánchez Luis Manuel
Ruíz Aguilar Cristian Jair
Gaytán Herrera Belén

Junio 2023

1 Introduction

Hoy en día, la comunicación a través de los sistemas informáticos se ha convertido en un factor prioritario para el desarrollo humano y de las sociedades: existen en las redes sociales, en los buscadores web, en los sitios web e-commerce, en softwares de desarrollo, etc. Para lograr esto, tanto personas como empresas se ven en la necesidad de alquilar su espacio en la web y darse a conocer en Internet.

Los sitios web son una de las múltiples formas de darse a conocer y ofrecer servicios a los usuarios de forma electrónica, que consiste en una interfaz personalizada accesible desde la Internet, capaz de ofrecer algún servicio a un usuario. Pero para brindar una mejor experiencia y satisfacer las necesidades de los clientes, se requiere de al menos tres aspectos clave en el desarrollo de un sitio web: una capa de presentación (lo que el cliente ve), una capa de negocio (la interactividad del sitio web) y una capa de datos (un software capaz de almacenar datos para su futuro uso). Este último software es conocido como una base de datos.

El presente proyecto consiste en la construcción y el diseño de una base de datos, aplicando todos los principios básicos de desarrollo, para ser implementada en el sitio web de una cadena de papelerías que busca innovar en la forma en que almacena su información, consulta datos, registra eventos o extrae la misma información, según los requerimientos de la cadena.

2 Plan de trabajo

2.1 Requerimientos

A continuación se muestra el formato en que el cliente (es decir, la cadena de papelerías) acudió a nosotros y nos expresó lo que necesita:

"Diseñar una base de datos para una cadena de papelerías que busca innovar la manera en que almacena su información. Se contrata a un equipo de trabajadores para que desarrollen los sistemas informáticos para satisfacer los siguientes requerimientos: Se desea tener almacenados datos como la razón social, domicilio, nombre y teléfonos de los proveedores. RFC, nombre, domicilio y al menos un correo electrónico (e-mail) 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, foto (opcional), 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. Adicional al almacenamiento de información, se requiere que el sistema resuelva lo siguiente:

- De manera automática, se genere una vista que contenga información necesaria para asemejarse a una factura de una compra.
- 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.
- 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 altera. Debe actualizarse el total a pagar por artículo y el total a pagar por la venta.
- Permitir obtener el nombre de aquellos productos de los cuales hay menos de 3 en stock.
- Al recibir el código de barras de un producto, regrese la utilidad.
- Crear un índice que satisfaga una necesidad que pudiera surgir durante la implementación.
- Requerimiento opcional: Diseñar en algún lenguaje de programación una interfaz gráfica que sea accesible a través de una URL o una aplicación móvil, que permita:
  - Agregar la información de un cliente.
  - 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."

2.2 Análisis de las actividades a realizar

Cada requerimiento expresado por el cliente debe ser implementado de acuerdo a las características de integridad de las bases de datos, respetando cada una y asegurar su almacenamiento para su consulta.

Se dividieron cada uno de los requerimientos en diferentes conjuntos, los cuales comienzan desde la creación de un modelo entidad-relación (MER) para representar la existencia de los datos en la base de datos (Figura 1.), hasta la implementación en una interfaz gráfica (Figura 2.) Asimismo, se asignó a cada miembro del proyecto una tarea específica para completar la ruta del desarrollo de éste.

Figure 1: Diagrama de Gantt de la base de datos

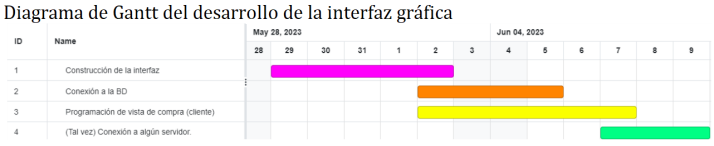


Figure 2: Diagrama de Gantt de la interfaz gráfica

Figure 3: Diagrama de Gantt de la documentación

En los diagramas de Gantt anteriores se observa una forma de trabajo en la que los miembros del equipo de trabajo completarán cada tarea de forma paralela, en un intervalo de tiempo que contempla dos semanas para la entrega del producto final. Una vez ponderados los tiempos y recursos requeridos por cada diagrama, en la Tabla 1 se muestra la distribución de integrantes del equipo a cada diagrama. La repetición de nombres en algunos diagramas contempla el

intercambio de puntos de vista para respetar los tiempos asignados para cada tarea.

Diagrama de Gantt 1								
Intergante	1	2	3	4	5	6	7	8
Mejia Ramos Bryan	x	x	x	x	-	x	x	x
García Sánchez Luis Manuel	x	x	x	-	-	-	-	-
Gaytán Herrera Belén	x	x	x	x	x	-	x	x
Ruíz Aguilar Cristian Jair	x	x	x	x	x	-	-	-

Diagrama de Gantt 2				
Intergante	1	2	3	4
Mejia Ramos Bryan	-	-	-	-
García Sánchez Luis Manuel	-	-	-	-
Gaytán Herrera Belén	-	-	-	-
Ruíz Aguilar Cristian Jair	x	x	-	-

Diagrama de Gantt 3			
Intergante	1	2	3
Mejia Ramos Bryan	x	x	x
García Sánchez Luis Manuel	-	-	-
Gaytán Herrera Belén	x	x	x
Ruíz Aguilar Cristian Jair	x	x	x

3 Diseño

Cada fase del diseño se analizó con detalle para mantener la integridad de la base de datos. A continuación se muestran los resultados resumidos del proceso de creación.

3.1 Creación de la base de datos

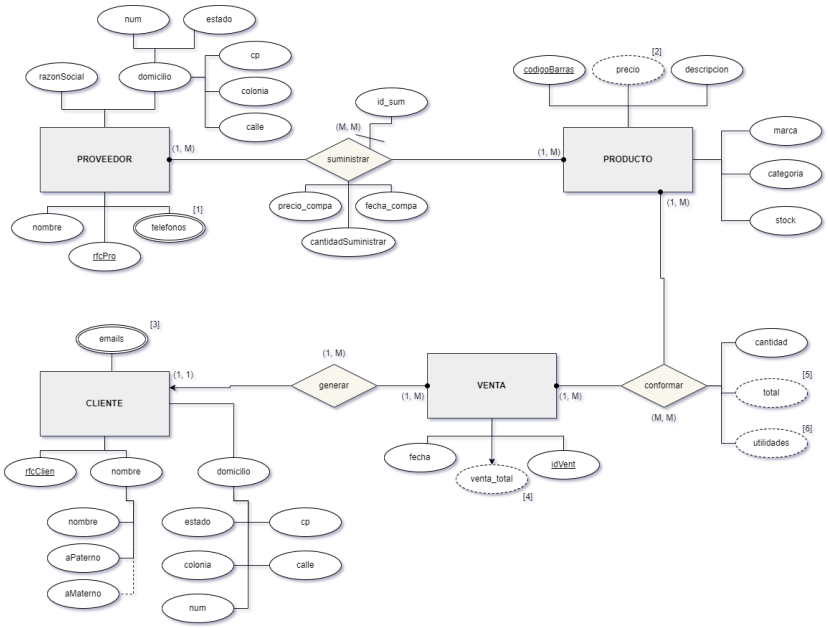


Figure 4: Modelo Entidad-Relación

Referencias:

[1]: Un proveedor puede registrar más de un teléfono. [2]: El valor del precio depende del atributo precio_compra de la relación suministrar. [3]: Los clientes deben tener al menos un email. [4]: La venta total se calcula con la suma de los totales parciales de la relación conformar. [5]: El total se calcula con el precio de los productos multiplicado por la cantidad comprada. [6]: Las utilidade se calculan de acuerdo a una tarifa programada, que multiplica el precio de la entidad producto por cierta cuantía.

3.2 Mapeo de las entidades a relaciones/tablas

Creación de tablas de entidades y creación de tablas de relaciones

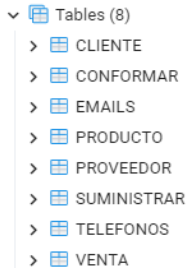


Figure 5: Tablas generadas en PostgreSQL.

TABLA PRODUCTO

Atributos: código de barras, precio, categoría, marca, descripción y cantidad en bodega (stock).

Se procuró que el precio de cada producto de la tabla PRODUCTO dependiera enteramente del valor de compra, es decir, del atributo precio_compra de la tabla SUMINISTRAR. Para tener un control en cuanto a una asignación de precios más íntegra y realista, se pueden programar tarifas predeterminadas para asignar precios de venta. Estas tarifas se asignaron, por default, como porcentajes extra que toman como referencia los precios de compra.

El código de barras de cada producto es generado a través de una concatenación: los 3 primeros caracteres son generados dependiendo el nombre de la categoría, los siguientes 3, dependiendo de la marca; los siguientes 3, dependiendo de la descripción; y finalmente un número genérico igual a 1000

El precio de los productos es suceptible a tener el valor de NULL, pues se entiende que aún no están a la venta. Una vez que se compran los productos, es decir, se registra su ingreso en la tabla SUMINISTRA, se asigna el precio a través de una función.

TABLA SUMINISTRAR

Atributos: ID de suministro, precio de compra, fecha de compra, cantidad suministrada y código de barras.

El ID de suministro es un número serial, el cual se programa como atributo único o candidato para mantener la unicidad de cada uno de los suministros. El precio de compra guarda el precio al que el producto fue comprado. La fecha de compra guarda la fecha de ingreso de los nuevos productos, la cual se genera con la función Now(). La cantidad suministrada es el número de piezas que adquirimos. Y el código de barras es el que se propagó de la tabla producto.

Nosotros queremos tales productos de tales categorías con tales marcas y se insertan en el proceso se crea su id, pero este no tendrá precio a menos que lo compremos y para ellos usamos suministrar, agregando el id del producto, el rfc del proveedor, el costo que nos oferto y la fecha. En base a esto podemos sacar la utilidad que usaremos en conformar para obtener la ganancia. En esta tabla nos encontramos con utilidad, el código de barras del producto que conforma la transacción, el id de la venta, y la cantidad de productos comprados que sirven para obtener el total usando el precio del producto.

TABLA CLIENTE

Atributos: RFC, nombre, apellido paterno, apellido materno, estado, colonia, código postal, calle, número.

El apellido materno del cliente puede ser opcional, pues no siempre es muy demandado para la documentación. El atributo nombre puede incluir tanto nombre de pila como nombres adicionales.

TABLA PROVEEDOR

Atributos: RFC, razón social, estado, colonia, código postal, calle, número.

Se reconoce de forma única a cada proveedor por el RFC de la empresa de la que proviene. Se decidió no optar por usar la razón social debido a su longitud.

TABLA EMAIL

Atributos: emails, RFC del cliente.

Los clientes pueden tener asociado a su RFC más de un email

TABLA TELEFONO

Atributos: telefonos, RFC del proveedor.

Los proveedores pueden tener más de un teléfono asociado a su RFC.

TABLA CONFORMAR

Atributos: cantidad, total y utilidades.

A través de esta tabla se tiene acceso al ID de una venta en específico y a los códigos de barras de los productos relacionados. La cantidad representa cuántos productos se compraron. El total, la multiplicación del precio unitario por la cantidad. Las utilidades representan el precio de venta menos el precio

de compra. Cabe mencionar que éstas son información un poco sensible.

TABLA VENTA

Atributos: ID de venta, fecha de venta y total de venta.

El ID de venta cuenta con la estructura "VENT-XXX", donde XXX representa un número serial que irá en aumento conforme se agreguen nuevas ventas. La fecha de venta registra la fecha en que se realizó la venta. El total de venta se calcula de la suma de los totales parciales de la tabla CONFORMAR

4 Implementación

4.1 Vista factura

La vista factura se logra a través de una función que recibe como parámetros dos valores: RFC del cliente y el ID de la venta. Es importante aclarar que si alguno de estos datos de esta dupla es erróneo o no existe, la función no puede proporcionar retroalimentación. En ella, a través del uso de cláusulas JOIN y bucles FOR, se logra reunir los datos que se consideraron más relevantes para la interfaz de una factura, así como información del emisor, que en este caso es la Papelería Clarky. El texto mostrado en la consola es retroalimentado gracias a la cláusula RAISE INFO.

Figure 6: Código factura.

4.2 Nombre de la venta

Se generó una secuencia para establecer el formato del ID de venta, permitiendo que sea ingresado y posteriormente llamado cuando sea necesario.

Figure 7: Código índice de venta.

4.3 Ganancia por periodo de tiempo

Para poder consultar de forma sencilla este aspecto, se utilizó una función que recibe por parámetro la fecha de la que se quiere obtener el total vendido y haciendo uso de una consulta se obtiene la suma total.

Figure 8: Código ganancia de fechas

4.4 Decremento de stock y advertencia

Con el objetivo de realizar el incremento en el stock, se implementó un trigger junto con su función correspondiente. Además, se incluyeron declaraciones condicionales (if) para validar diversos casos y asegurar que el incremento del stock se realice de manera adecuada.

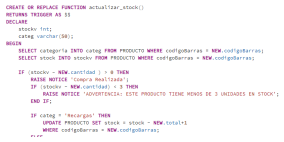


Figure 9: Código decremento de stock y advertencia.

4.5 Visualizar stock

Se utilizó una función que incluye una consulta para mostrar los productos con un stock inferior a 3.

4.6 Utilidad

A partir del código de barras, se obtiene la utilidad correspondiente en función de la categoría del producto.

La funcion realiza un if para determinar la categoria y en base a eso en impresiones las que sean blanco y negro costaran 60 porciento mas mientras que las de color costraran el doble que las de b/N por lo que la utilidad es el 60 porciento y en tal caso las de color el 120 porciento, los regalos la utilidad es el 10 porciento, los articulos de papeleria el 85 porciento y las recargas tendran la utilidad de 1, este valor es el que se devuelve en caso de no pertener devuelve 0.

Figure 10: Código de la utilidad de un producto dado.

4.7 Precio del producto

A partir del precio de costo al que se vendio a partir del join con suministrar este devolverá el precio el cual es la suma de el costo de suministro + la utilidad.

Figure 11: Código del precio.

4.8 Actualizar_stock

La función forma parte de un TRIGGER que monitorea las inserciones de la tabla CONFORMAR, cada que se realiza una inserción verifica que hay suficiente stock para realizar la compra, de lo contrario cancela la inserción en la tabla CONFORMAR y no actualiza nada, si hay el suficiente stock, actualiza el stock del producto correspondiente, verifica mediante un if si se trata de una Recarga o un otra categoría, si estamos en el primer caso, resta diferente el stock (el stock representa el saldo disponible para las recargas) y permite la inserción del registro en la tabla CONFORMAR.

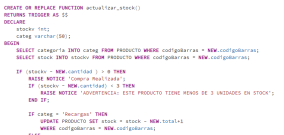


Figure 12: Código de la actualización de los precios.

4.9 ID Venta

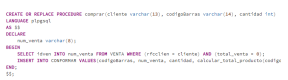
Esta función contiene una consulta inserta en la tabla VENTA una venta con Idven calculado por la función id_venta_serial(), pero poniendo el valor total de la venta en 0 para indicar que se trata de una venta en proceso.

```
CREATE SEQUENCE id_venta_secuencial;

ALTER SEQUENCE id_venta_secuencial START 1 INCREMENT 1 NO MAXVALUE MINVALUE 1;
--Esta sentencia hace que empiece en 1, vaya aumentando de a 1, no haya límite en el 001 y el mín

CREATE OR REPLACE FUNCTION id_venta_secial() --crear la función
RETURNS TEXT AS

$$
DECLARE
  secuencia TEXT;

    BEGIN
        secuencia := 'VENT-' || LPAD(nextval('id_venta_secuencial')::TEXT, 3, '0'); --Se construc
        RETURN secuencia;
    END;

$$ LANGUAGE plpgsql;
```

Figure 13: Código del ID de venta VENT-001

4.10 Comprar

Esta función recibe por parámetros el rfc del cliente, el código de barras del producto y la cantidad que se quiere comprar de dicho producto, después con una consulta va insertando los registros en la tabla CONFORMAR haciendo uso del idVen que fue generado con venta() al inciar la venta del cliente.

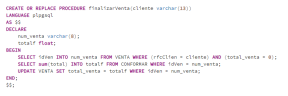


```
CREATE OR REPLACE PROCEDURE comprar(cliente varchar(13), codfgdbarras varchar(14), cantidad int)
LANGUAGE plpgsql
AS $$
DECLARE
    num_venta varchar(8);
BEGIN
    SELECT idven INTO num_venta FROM VENTA WHERE (rfcclien = cliente) AND (total_venta = 0);
    INSERT INTO CONFORMAR VALUES(codfgdbarras, num_venta, cantidad, calcular_total_producto(codfgb
END;
$$;
```

Figure 14: Código de la realización de una compra.

4.11 Finalizar una venta

Esta función lo que hace es concluir con la venta que se realizó, al utilizar las funciones venta(), comprar(), recopila todos los productos pertenecientes a la venta actual , suma sus precios y actualiza el campo de total de venta en la tabla VENTA con el agregado final del precio de todos lo productos.



```
CREATE OR REPLACE PROCEDURE ffinalizarVenta(cliente varchar(13))
LANGUAGE plpgsql
AS $$
DECLARE
    num_venta varchar(8);
    totalf float;
BEGIN
    SELECT idven INTO num_venta FROM VENTA WHERE (rfcclien = cliente) AND (total_venta = 0);
    SELECT sum(total) INTO totalf FROM CONFORMAR WHERE idven = num_venta;
    UPDATE VENTA SET total_venta = totalf WHERE idven = num_venta;
END;
$$;
```

Figure 15: Código del cierre de una venta.

4.12 Calcular_total_producto

Esta función utiliza el código de barras de cada producto para realizar una consulta y obtener el precio correspondiente. El resultado de la función es el precio del producto.

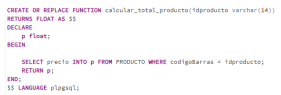


Figure 16: Código del cálculo total de un producto.

4.13 Calcular la ganancia de una fecha

Esta función realiza una consulta con una función de agregación en la tabla "VENTA" para obtener el total obtenido de las ventas por día. El resultado es el monto total de ventas registrado para cada día.

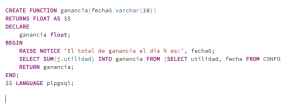


Figure 17: Código de la ganancia de un día.

5 Presentación

Nuestro equipo de trabajo considera que el estado de la interfaz gráfica del sitio web, junto con la base de datos, debe mantenerse aún bajo pruebas controladas para evitar cualquier tipo de inconveniente o inconsistencia al momento de registrar y extraer datos.

Para que el cliente pueda verificar, sin embargo, el buen funcionamiento de la base de datos, se sugiere la instalación de un software especializado en el manejo de ésta (DBMS, por sus siglas en inglés) o la asistencia de algún miembro del equipo en todo momento para realizar cualquier tipo de consultas.

A continuación se muestran algunas imágenes de los resultados de la interfaz gráfica aún en proceso de pruebas.

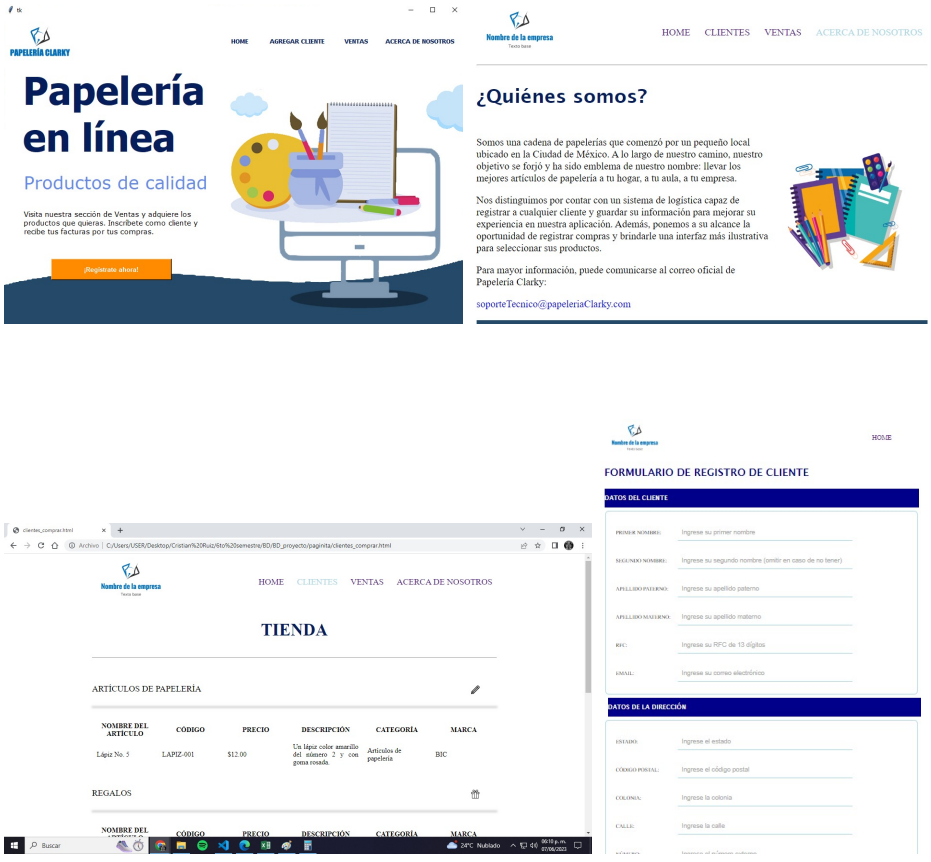


Figure 18: Capturas de pantalla de la interfaz gráfica (UI)

6 Conclusiones

Mejia Ramos Bryan
Creo que el proyecto fue una gran forma de poner todos los conocimientos en su lugar, ya que al implementar algo tan completo y desde 0 hizo que aprendiera todo de nuevo, por ello creo que fue un proyecto del que aprendí mucho y que se me hizo muy interesante, presentó demasiados retos, tanto logísticos como implementación, pero creo que de ese tipo de dificultades es de las que más se aprende, por ello puedo concluir que fue un gran cierre de curso porque fue muy complementario con para finalizar los aprendizajes.

Ruíz Aguilar Cristian Jair
El proyecto fue una manera masiva y dura de poner en práctica todos los conocimientos adquiridos en el tiempo tan exacto para entender los temas de aplicación. Requerí de mayor estudio y mayor práctica de la programación de consultas, pues a mi parecer, este tema debió haberse practicado como un programador resuelve problemas en cualquier lenguaje de programación. También me ayudó a reforzar mis ideologías del trabajo en equipo y reafirmar mis decisiones para mi desarrollo como profesional que contribuya en proyectos grupales (que a largo plazo probablemente serán muchos).

Gaytán Herrera Belén
el proyecto ayudó mucho a reforzar conocimientos adquiridos a lo largo del semestre poniendo en práctica el crear tablas y acceder a los datos con joins realizar comparaciones y operaciones, lo que mas me costo fue saber lo que quería en los joins creo que con tiempo se podian lograr más cosas la función que me costó más trabajo fue la transformación de datos para obtener el precio en las recargas aprendí a transformar datos varchar a integer de forma directa, así mismo aprendí a concatenar por solo algunas letras como lo es el código de barras con uso de string una funcion muy util.