Papeleria Grafito's co.

Suárez Román Clara Alin
Tapia Solís Samuel Alejandro

I. Índice

1.Objetivo
2.Introducción
3.Plan de trabajo
4.Diseño
5.Implementación
6.Presentación
7.Conclusiones

II. Objetivo

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

III. Introducción

Hace tiempo, veíamos que en lugares grandes como empresas o algo pequeño como un consultorio médico toda la información que se tenía registrada era guardada en folders o carpetas para llevar un control de toda la información del lugar, hoy en dia es muy poco común que nos encontremos con algo así, ya que existen plataformas o lenguajes que nos ayudan para llevar el control de toda esa información, a toda esa información guardada y organizada almacenada en un sistema electrónico lo llamamos base de datos.
Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos.
Los datos dentro de los tipos más comunes de bases de datos en funcionamiento hoy en día se modelan típicamente en filas y columnas en una serie de tablas para que el procesamiento y la consulta de datos sean eficientes. Luego se puede acceder, administrar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan lenguaje de consulta estructurado (SQL) para escribir y consultar datos.

Pero la gestión de una base de datos en algún lenguaje es solamente lo que se llama "backend", lo que quiere decir que solamente es información que está en un dispositivo guardada pero como tal no podemos visualizarla a menos que sea con una interfaz que nos permita hacerlo, para darle solución a esta problemática, tenemos al "frontend" el cual nos va a servir para que el usuario pueda consultar toda la información que necesite ya sea a través de una app móvil o una página web, por poner un ejemplo, para ello es necesario tener en cuenta que para poder unir nuestra página web con la base de datos, es necesario un lenguaje, en este caso hablaremos de php el cual nos va a permitir conectar nuestra base de datos con nuestra página web.

IV. Plan de trabajo

Actividades a realizar:

Diseño conceptual

-Leer los requerimientos: en esta parte el equipo del proyecto se encargará de leer los requerimientos del proyecto para poder comprender lo que se está pidiendo.
-Realizar el modelo de entidad relación: se identificará las entidades, así como los atributos necesarios para la creación de la base de datos.

Diseño Lógico

-Realizar el modelo relacional: una vez obtenido el MER, se realizará el modelo relacional de las entidades ya establecidas anteriormente.

Diseño Físico

-Programación en postgresql: ya que tenemos el modelo relacional se iniciará la programación en el lenguaje escogido que en este caso es postgresql. -Creación de la página web: una vez creada la base de datos se creará la página web utilizando el lenguaje html, javaScript, CSS.
- Unir ambos códigos: por último ya que tengamos ambas partes del proyecto, es decir, la gráfica como la base de datos; se utilizará el lenguaje php para poder unir ambas partes.

El cronograma de actividades se dividió en 4 semanas con las correspondientes fechas:
Semana 1: 28/12/2020 - 03/01/2021
Semana 2: 04/01/2021 - 10/01/2021
Semana 3: 11/01/2021 - 17/01/2021
Semana 4: 18/01/2021 - 24/01/2021

Cronograma de actividades					
Miembro del equipo	Actividad a realizar	Semana 1	Semana 2	Semana 3	Semana 4
Tapia Solis Suárez Román	Requerimientos				
Tapia Solis Suárez Román	MER				
Tapia Solis Suárez Román	MR				
Tapia Solis Suárez Román	Programación Postgresql				
Tapia Solis Suárez Román	Programación Html				
Tapia Solis	Programación Php				
Suárez Román	Documentación				

Hình 1. Imagen 1. Cronograma de actividades

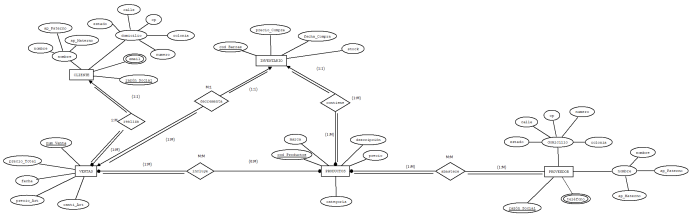
V. Diseño

Diseño conceptual

En esta parte del proyecto nos dedicamos a entender lo que se pide en el proyecto y así poder crear las entidades con sus atributos necesarios para que se forme una base de datos completa y sólida.

Modelo entidad relación

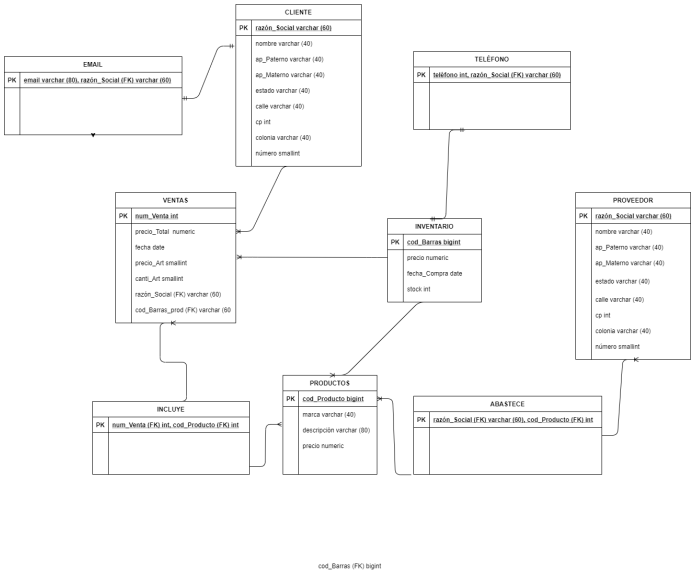
Una vez leído el enunciado del proyecto comenzamos la creación del diagrama de entidad relación.
Nosotros identificamos 5 entidades inicialmente: VENTAS, PRODUCTOS, CLIENTE, INVENTARIO, PROVEEDOR.
Cada una con sus atributos correspondientes como lo muestra el diagrama que se hizo en la aplicación Día..



Hình 2. Imagen 2.Modelo Entidad Relación

Diseño Lógico

Una vez teniendo el modelo de entidad relación, lo siguiente que hicimos fue crear el modelo relacional, ocupando las reglas vistas en clase.
A cada entidad la convertimos en una relación y a los atributos les agregamos el tipo de atributo así como su longitud en caso de ser necesario.
Los atributos que son multivaluados los hicimos una relación independiente donde incluimos la llave primaria como llave foránea de la nueva relación. En este caso sólo tuvimos dos atributos multivaluados: teléfono e e-mail.
En el MR las relaciones cuando son de M:M se crea otra nueva entidad con las llaves primarias de las entidades que se relacionan y en el caso que sea M:1 o viceversa entonces se pasaría como llave foránea la llave primaria de una de las entidades.
En la imagen se puede observar el diseño de las relaciones.



Hình 3. Imagen 3.Modelo Relacional

Diseño Físico

Programación en postgresql

Ya que se tiene el modelo relacional, lo siguiente que se hace es empezar a crear las tablas. Nosotros lo hicimos a través de la herramienta de PostgreSQL “pgAdmin”.

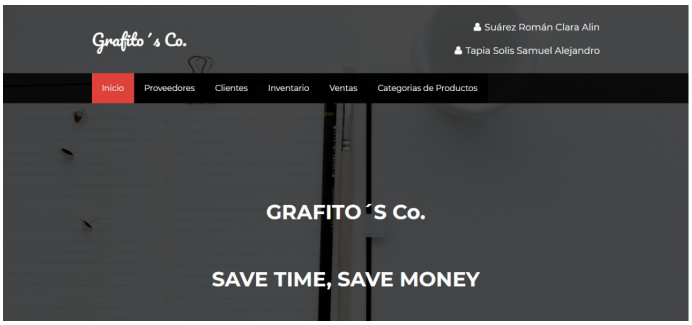
```
-- Creación de tablas

create table cliente(
    razon_Social varchar(60),
    nombre varchar(40),
    ap_Paterno varchar(40),
    ap_Materno varchar(40),
    estado varchar(40),
    calle varchar(40),
    cp int,
    colonia varchar(40),
    numero smallint,
    primary key(razon_Social)
);
```

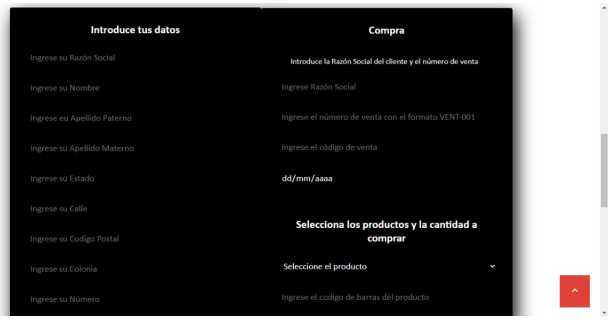
Hình 4. Imagen 4. Modelo Relacional

Página web

Para la página web utilizamos 3 lenguajes los cuales fueron: html, JavaScript y CSS.



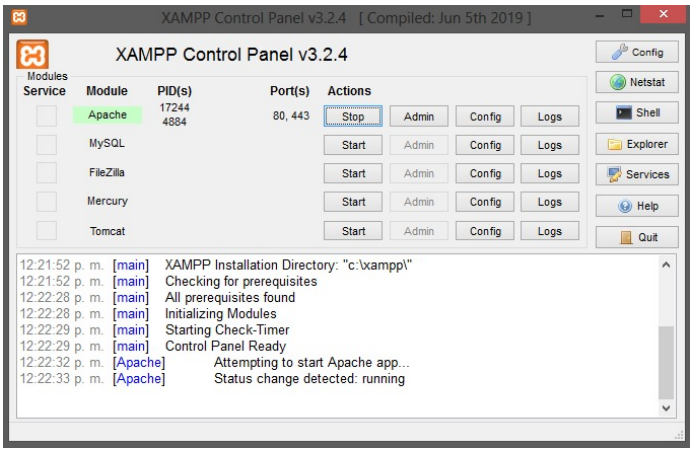
Hình 5. Imagen 5. Página wb



Hình 6. Imagen 6. Página web

Unión de los códigos

Para poder enlazar la bse de datos con la página web ya creada se utilizó el lenguaje Php pero además utilizamos XAMPP que es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.



Hình 7. Imagen 7. Página web

VI. Implementación

1. Al recibir un código de barras de un producto, regrese su utilidad.

Para obtener la utilidad del producto se necesita hacer una resta entre el precio al que se vende el cual se encuentra en la tabla “productos” y el precio al que se compra el cual se encuentra en la tabla ”inventario”.
Para obtener estos datos de las tablas se utilizó un full outer join, ya que con este join se puede obtener todos los datos de ambas tablas.
El elemento en común que utilizamos fue el código de barras que es la llave primaria de inventario pero también es la llave foránea de productos.
Una vez que se obtienen los datos sólo se realiza la resta

```
-- 1. Devolver la utilidad

select cod_barras,descripcion,precio_compra,precio,'La utilidad es:', precio - precio_compra
from inventario as inv full outer join productos as pro
on inv.cod_barras=pro.cod_barras_inventario
where cod_barras = 736547265378
```

Hình 8. Imagen 8. Utilidad

2. 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.

Se crea una función, porque siempre que se quiera ejecutar un trigger se tiene que hacer una función, y ponemos “returns

trigger” para decirle que queremos que en esa función nos regrese un trigger.
Después dentro de esa función va lo que queremos que haga el trigger, en este caso va a decrementar dependiendo lo que se haya introducido en la compra al stock.
El where es para decirle que eso se va a hacer cuando el código de barras de la compra introducida sea igual a el código de barras del inventario.

```
-- 2. Decrementarse stock

create function actualiza_stock() returns trigger as
$$
begin

update inventario set stock=stock - producto_vendido.cantidad_art
where producto_vendido.cod_barras_inventario = inventario.cod_barras;

return new;
end;
$$ language plpgsql;
```

Hình 9. Imagen 9. Decrementar el stock

Y en esta parte estamos creando como tal el trigger, y estamos diciendo que ese trigger se va a hacer cuando se haga una función insert en la tabla de “producto_vendido”.
Y mandamos a llamar la función en donde está el trigger que en este caso se llama actualiza_stock.

```
create trigger TR_insert before insert on producto_vendido
for each row
execute procedure actualiza_stock();
```

Hình 10. Imagen 10. Decrementar el stock

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

Para realizar esta operación seleccionamos el atributo “precio_total” el cual se encuentra en la tabla ventas y utilizamos la función “sum” la cual se encargará de realizar la suma donde se utiliza un between para establecer las fechas entre las cuales se quiere realizar la suma.

```
-- 3. Dada una fecha devolver la cantidad que se vendio

select sum(precio_total) from ventas
where fecha between '2021-01-01' and '2021-02-01'
```

Hình 11. Imagen 11. Cantidad total vendida

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

Para este punto utilizamos un inner join entre la tabla “inventario” y la tabla “productos” para obtener el nombre del producto y el stock. El elemento en común que utilizamos fue el código de barras que es llave primaria de inventario y llave foránea de productos. Una vez obteniendo estos datos ponemos una condición “where” donde indique que se van a mostrar los datos del select siempre y cuando el stock sea menor a tres.

```
-- 4. Nombre de productos de los cuales hay 3 en stock

select descripcion,stock from productos as pro inner join inventario as inv
on pro.cod_barras_inventario=inv.cod_barras
where stock <=3
```

Hình 12. Imagen 12.Stock menor a 3

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

Para obtener todos los datos tuvimos que hacer dos joins para unir los datos de tres tablas: ventas, producto_vendido y productos. El primero join es entre ventas y producto_vendido donde el elemento en común es el número de venta que es la llave primaria de la tabla venta pero también es la llave foránea de la tabla producto_vendido.
Una vez que se obtuvieron esos datos hacemos el siguiente join con la tabla productos donde el elemento en común es el código de barras de inventario.
Por último le agregamos la condición where para que se muestre cuando se haga la compra, es decir, a partir del número de venta.

```
-- 5.

select fecha,num_venta,razon_social_cliente,descripcion,precio_art,cantidad_art,precio_total
from ventas as ven join producto_vendido as pv
on ven.num_venta=pv.num_venta_ventas
join productos as prod
on prod.cod_barras_inventario=pv.cod_barras_inventario
where num_venta=1
```

Hình 13. Imagen 13. Factura de compra

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

Nosotros elegimos crear un índice en la tabla productos ya que nuestra tablas contiene todas las categorías juntas, es decir; papelería, recargas, impresiones y regalos. Entonces para tener la tabla un poco más ordenada creamos un índice en el atributo “categoría” para que se ordene alfabéticamente .

```
-- 6.

create index indice_producto on productos (categoria);
create index indice_prod on productos (cod_productos);
```

Hình 14. Imagen 14.índice

VII. Conclusiones

Tapia Solís Samuel Alejandro

En lo personal nunca había trabajado con la gestión de una base de datos y menos que estuviera ligada a una página web, solo había trabajado por aparte una página web y una base de datos, y pensaba la unión de ambas era algo relativamente sencillo como indicarle a la página web que tenga algún movimiento mediante JavaScript, pero me di cuenta que nada que ver, es muy complicado hacer que estén ligadas, pero me ayudo mucho para entender mejor todo lo que se vio en el curso, ya que se aplico de todas las maneras posibles lo visto, me quedo con una muy buena experiencia.

Suárez Román Clara Alin

Creo que este proyecto presentó muchas dificultades para mí ya que si bien sí aprendí mucho en la materia de teoría creo que lo que no aprendí muy bien fue cómo aplicar los conceptos ya en la programación, con esto me refiero a que yo sé cómo pasar del MER al MR y también sé crear las tablas en postgresql pero al momento de trabajar con ellas creo que fue lo que me falló. Otro punto que se me complicó un poco fue el poder ligar la base de datos con la página web. Pero creo que más que se me complicara fue que no le di el tiempo suficiente para poder investigar todas mis dudas. Pero aunque el proyecto no fue lo esperado , me di cuenta que la creación de las bases de datos es algo que me llama mucho la atención y tal vez sea la rama en la cual quiera especializarme.