

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**BASES DE DATOS**

**SISTEMA DE COMERCIO ELECTRÓNICO**

**22 de Junio de 2021**

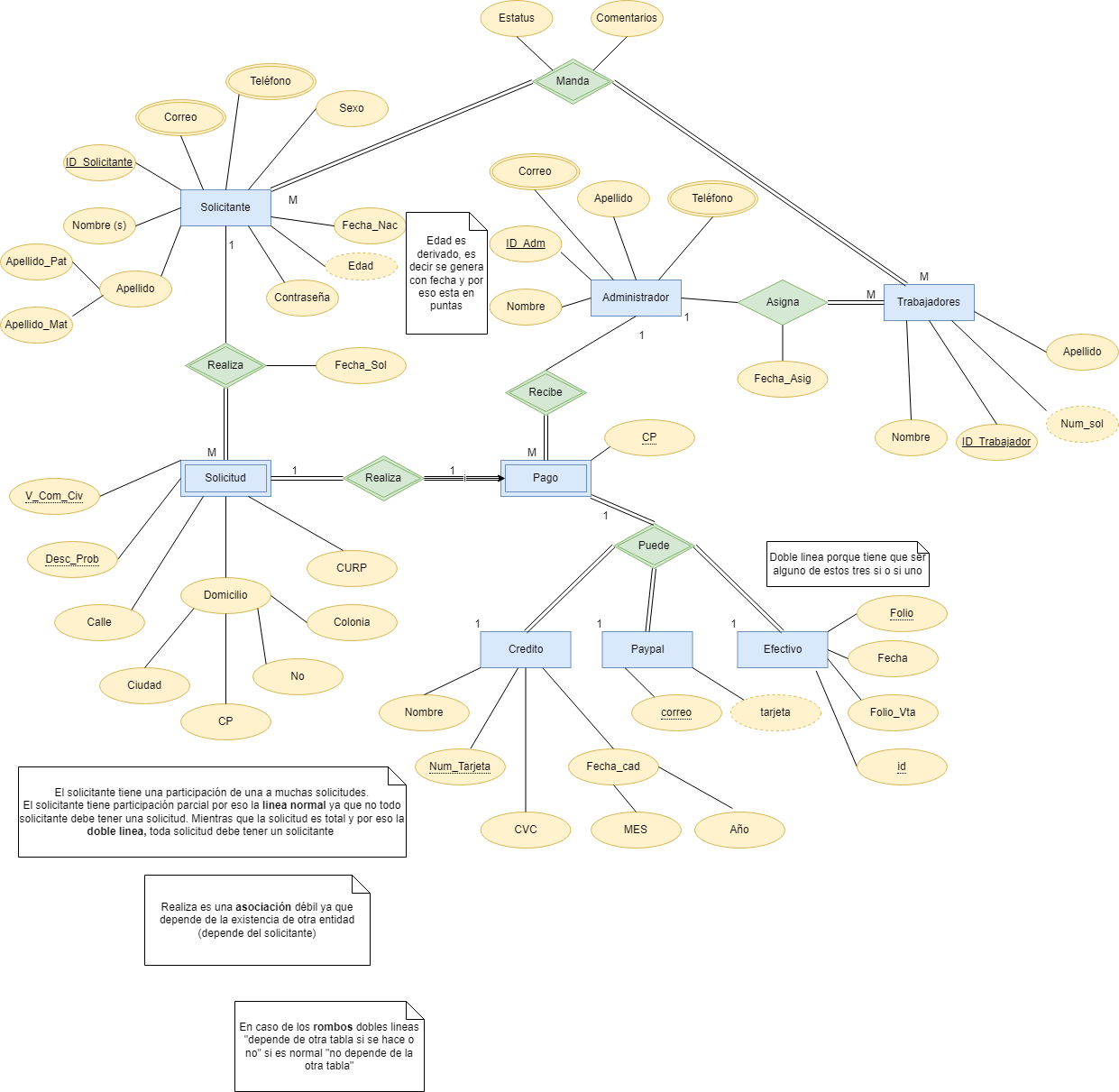
**2. Descripción general del proyecto**

El proyecto consiste en una base de datos para un sistema de comercio electrónico, la cual será construida a partir de las tecnologías y herramientas actuales, el sistema de comercio electrónico estará basado en la venta de refacciones de automóviles, de acuerdo a esto, las reglas del negocio tendrán características específicas que se describirán en el apartado correspondiente.

* Es un despacho de abogados que quiere automatizar las demandas de sus clientes, esto lo harán a traves de una página web llenando un formulario.
* Al momento de llenar el formulario se manda al proceso de pago para finalizar la transacción.
* Para dar seguimiento a su demanda, el cliente crea una cuenta en la plataforma y verá el seguimiento de cada una de las actualizaciones del proceso legal.
* El administrador del sitio recbe la notificación de una nueva demanda y con los datos llenados del formulario se crea automaticamente el documento legal en formato word para empezar el proceso.
* El administrador recibe el pago y debe de ser capaz de verlo en un dashboard para ver la cantidad de ingresos recibidos.
* El administrador actualiza el proceso de la demanda y agrega comentarios en cada paso del proceso.
* Al usuario le llegan correos de notificación para saber el avance de su proceso.
* La página debe de ser responsive para poderla ver desde el celular.
* La preferncia de colores del cliente es azul marino y blanco, pero acepta propuestas.

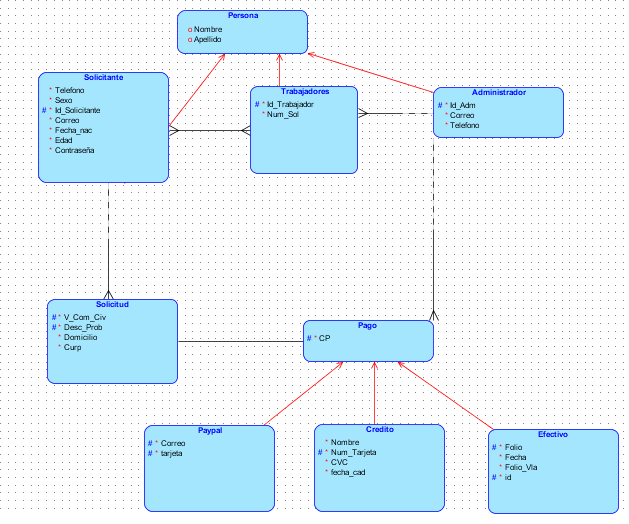
**3. Reglas del negocio**

1. Por usuario solo se permiten 3 solicitudes por usuario registrado.
2. Los clientes deben ser mayores de 18 años.
3. El solicitante una vez procese el pago se considera una solicitud
4. Los clientes por ser nuevos reciben un 5% de descuento por su solicitud fiscal completada.
5. Los clientes que ya es su segunda solicitud reciben el 15% de descuento en la solicitud completada.
6. El despacho antes de 3 días de su fecha limite deben mandar su comentario sobre el proceso legal, es decir que cada dos semanas se deben mandar los comentarios o cambiaran de estado como “retraso”, si es dentro del límite “enviada”.
7. Los únicos medios de pago para las compras son con tarjeta (crédito o débito) o PayPal.
8. Hay fechas; inicio de proceso, en proceso, proceso concluido.
9. En el formulario de registro todos los datos son obligatorios
10. Cuando se registra se le asignara un id automáticamente (aplica para despacho y solicitantes, y algunos de administración del sistema).
11. El precio de la solicitud será asignada por tiempo, y estos proporcionados por el despacho.
12. A cada persona del despacho tendrá una carga de trabajo no mayor a 5, por lo que se balanceara la carga.
13. Se registrara un método de pago.

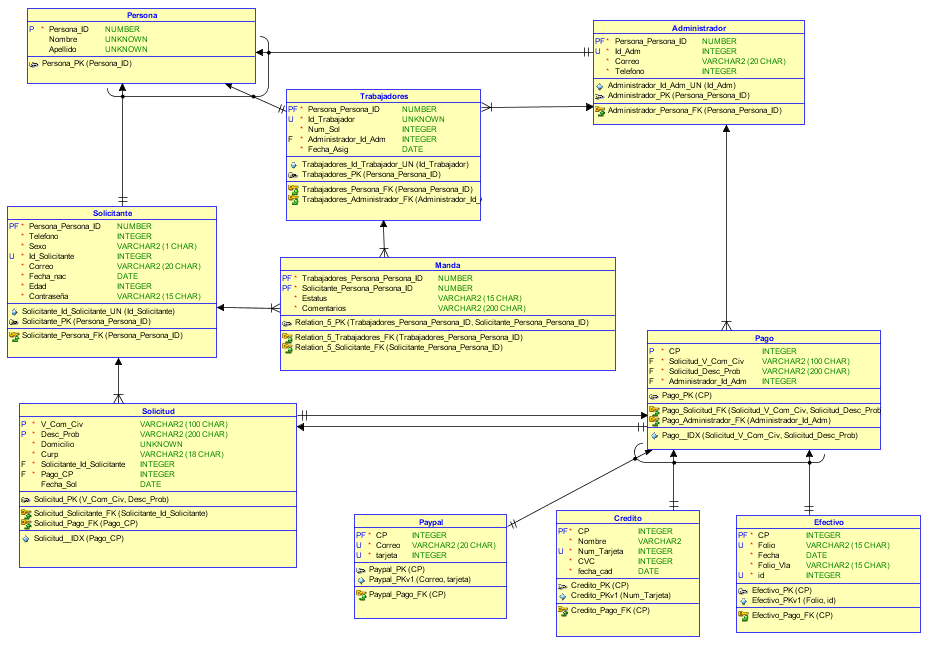


**4. Modelo Entidad Relación y relacional**

**Modelo Entidad relación**

****

**Modelo Relacional**



5. Código SQL de creación de objetos y restricciones

CREATE TABLE cliente (

  numusuario INTEGER NOT NULL PRIMARY key,

  nombre varchar2(30) NOT NULL,

  apellido\_pat varchar2(20) NOT NULL,

  sexo CHAR(1 CHAR) NOT NULL CHECK(sexo IN ('M', 'F') ),

  fecha\_nacimiento DATE NOT NULL,

  tipo varchar2(20) DEFAULT 'normal' NOT NULL,

  correo varchar2(20 CHAR) NOT NULL UNIQUE,

  telefono INTEGER

);

Alter table cliente add constraint valida\_edad check (extract(year from fecha\_nacimiento) < 2002);

Alter table cliente add constraint valida\_tipo CHECK(tipo IN ('repartidor', 'normal'));

CREATE TABLE producto (

  codigo INTEGER NOT NULL PRIMARY key,

  descripcion varchar2(50) NOT NULL,

  precio number(8, 2) NOT NULL,

  cantidad INTEGER NOT NULL,

  categoria varchar2(50) NOT NULL

);

CREATE TABLE favorito (

  cliente\_num INTEGER NOT NULL,

  producto\_codigo INTEGER NOT NULL,

  PRIMARY key(cliente\_num, producto\_codigo),

  FOREIGN key (cliente\_num) REFERENCES cliente(numusuario) ON DELETE CASCADE,

  FOREIGN key (producto\_codigo) REFERENCES producto(codigo) ON DELETE CASCADE

);

CREATE TABLE compra (

  id\_compra INTEGER NOT NULL PRIMARY key,

  fecha DATE DEFAULT sysdate NOT NULL,

  subtotal number(8, 2) NOT NULL,

  mediopago varchar2(10 CHAR) DEFAULT 'efectivo' NOT NULL,

  cliente\_num INTEGER NOT NULL,

  FOREIGN key (cliente\_num) REFERENCES cliente(numusuario) ON DELETE CASCADE

);

Alter table compra add constraint valida\_mediop CHECK(mediopago IN ('efectivo', 'paypal'));

CREATE TABLE contiene (

  producto\_codigo INTEGER NOT NULL,

  compra\_id INTEGER NOT NULL,

  cantidad INTEGER NOT NULL,

  PRIMARY key(producto\_codigo, compra\_id),

  FOREIGN key (producto\_codigo) REFERENCES producto(codigo) ON DELETE CASCADE,

  FOREIGN key (compra\_id) REFERENCES compra(id\_compra) ON DELETE CASCADE

);

Alter table contiene add constraint valida\_cantidad CHECK( cantidad >= 1 AND cantidad <= 20 );

CREATE TABLE repartidor(

  numero\_repartidor INTEGER NOT NULL PRIMARY key,

  nombre varchar2(30) NOT NULL,

  apellidop varchar2(30) NOT NULL,

  num\_telefono INTEGER NOT NULL,

  num\_unidad INTEGER NOT NULL UNIQUE

);

CREATE TABLE envio (

  numero\_envio INTEGER NOT NULL PRIMARY key,

  fecha\_salida DATE DEFAULT sysdate NOT NULL,

  fecha\_entrega DATE NOT NULL,

  delegacion varchar2(20 CHAR) NOT NULL,

  colonia varchar2(20 CHAR) NOT NULL,

  calle varchar2(25 CHAR) NOT NULL,

  numero INTEGER,

  referencias varchar2(100 CHAR) NOT NULL,

  compra\_id INTEGER NOT NULL UNIQUE,

  repartidor\_numero INTEGER,

  FOREIGN key (compra\_id) REFERENCES compra(id\_compra) ON DELETE CASCADE,

  FOREIGN key (repartidor\_numero) REFERENCES repartidor(numero\_repartidor) ON DELETE CASCADE

);

Alter table envio add constraint valida\_fechaen CHECK((fecha\_entrega - fecha\_salida) <= 5);

CREATE sequence no\_envio increment BY 1 START WITH 1 nomaxvalue minvalue 0;

CREATE OR replace PROCEDURE asigna\_rep (num\_env INTEGER) IS

 rep\_id repartidor.numero\_repartidor % type;

BEGIN

    ELECT numero\_repartidor INTO rep\_id FROM (SELECT numero\_repartidor FROM repartidor ORDER BY dbms\_random.VALUE) WHERE rownum = 1;

    UPDATE envio SET repartidor\_numero = rep\_id WHERE

    numero\_envio = num\_env;

    IF SQL % rowcount = 0 THEN

        dbms\_output.put\_line('No se encontró el envío, no se asignó repartidor');

    ELSE

    dbms\_output.put\_line('Se asignó el repartidor ' || rep\_id || ' al envio número ' || num\_env);

    END IF;

END;

CREATE OR replace PROCEDURE compra\_prod(num\_compra INTEGER, prod\_cod INTEGER, cant INTEGER) IS

 prod\_cant producto.cantidad % type;

BEGIN

    SELECT cantidad INTO prod\_cant FROM producto WHERE codigo = prod\_cod;

    IF prod\_cant > cant AND prod\_cant - cant > 0 THEN

    INSERT INTO contiene VALUES (prod\_cod, num\_compra, cant);

    UPDATE producto SETcantidad = cantidad - cant WHERE codigo = prod\_cod;

    ELSE

    dbms\_output.put\_line('No hay suficientes productos en existencia');

    END IF;

END;

create view detalle\_compra as select c.compra\_id, p.codigo, p.descripcion, p.precio, c.cantidad from producto p join contiene c on p.codigo = c.producto\_codigo;

declare

compraID compra.id\_compra % type:= 22;

compra\_sub compra.subtotal % type;

compra\_tot compra.subtotal % type;

compra\_fecha compra.fecha % type;

compra\_pago compra.mediopago % type;

 us\_tipo cliente.tipo % type;

 us\_nombre cliente.nombre % type;

 us\_ap cliente.apellido\_pat % type;

BEGIN

    SELECT c.tipo, c.nombre, c.apellido\_pat, co.fecha, co.mediopago INTO us\_tipo, us\_nombre, us\_ap, compra\_fecha, compra\_pago FROM cliente c join compra co on c.numusuario = co.cliente\_num  where co.id\_compra = compraID ;

    IF SQL % rowcount = 0 THEN

    dbms\_output.put\_line('No se encontró el usuario');

    ELSE

        SELECT SUM(precio) INTO compra\_sub FROM producto WHERE codigo IN (SELECT producto\_codigo FROM contiene WHERE compra\_id = compraID );

        IF SQL % rowcount = 0 THEN

        dbms\_output.put\_line('No se encontraron productos en la compra');

        ELSE

        if us\_tipo = 'repartidor' then

            compra\_tot := compra\_sub - (compra\_sub \* 0.07);

        else

        compra\_tot := compra\_sub ;

        end if;

            dbms\_output.put\_line('FECHA COMPRA: ' || compra\_fecha);

            dbms\_output.put\_line('COMPRA ID: ' || compraID);

            dbms\_output.put\_line('NOMBRE: ' || us\_nombre);

            dbms\_output.put\_line('APELLIDO: ' || us\_ap);

            dbms\_output.put\_line('SUBTOTAL: ' || compra\_sub);

            dbms\_output.put\_line('TOTAL: ' || compra\_tot);

            dbms\_output.put\_line('MEDIO DE PAGO: ' || compra\_pago);

        END IF;

    END IF;

end;

select \* from detalle\_compra where compra\_id = 22;

6. Código SQL de consultas de las reglas del negocio

1. Solo se permite hasta un máximo de 20 artículos de un solo producto por compra y un mínimo de 1.

Alter table contiene add constraint valida\_cantidad CHECK( cantidad >= 1 AND cantidad <= 20 );

select \*from user\_constraints where constraint\_name = ‘VALIDA\_CANTIDAD’

1. Los clientes deben ser mayores de 18 años.

Alter table cliente add constraint valida\_edad check (extract(year from fecha\_nacimiento) < 2002);

select \*from user\_constraints where constraint\_name = 'VALIDA\_EDAD'

1. Un cliente puede ser de tipo ‘Repartidor’ o ‘Normal.

Alter table cliente add constraint valida\_tipo CHECK(tipo IN ('repartidor', 'normal'));

select \*from user\_constraints where constraint\_name = 'VALIDA\_TIPO’

1. Se debe revisar que el número de productos del pedido sea menor que el número de productos en existencia.

create or replace  procedure compra\_prod(num\_compra INTEGER, prod\_cod INTEGER, cant INTEGER) IS

prod\_cant producto.cantidad%TYPE;

begin

select cantidad into prod\_cant from producto where codigo = prod\_cod;

if prod\_cant > cant and prod\_cant - cant > 0 then

insert into contiene values(prod\_cod, num\_compra, cant);

update producto set cantidad = cantidad - cant where codigo = prod\_cod;

else

dbms\_output.put\_line('No hay suficientes productos en existencia');

end if;

end;

select \*from user\_procedures where object\_name = 'COMPRA\_PROD';

1. Los únicos medios de pago para las compras son con tarjeta (crédito o débito) o PayPal.

Alter table compra add constraint valida\_mediop CHECK(mediopago IN ('efectivo', 'paypal'));

select \*from user\_constraints where constraint\_name = 'VALIDA\_MEDIOP’

1. La fecha de entrega debe estar dentro de un plazo de 5 días después de la fecha de salida.

Alter table envio add constraint valida\_fechaen CHECK((fecha\_entrega - fecha\_salida) <= 5);

select \*from user\_constraints where constraint\_name = 'VALIDA\_FECHAEN’

1. A cada envío se le debe asignar un repartidor.

create or replace procedure asigna\_rep (num\_env INTEGER) IS

rep\_id repartidor.numero\_repartidor%TYPE;

BEGIN

    SELECT Numero\_repartidor INTO rep\_id FROM (SELECT Numero\_repartidor FROM repartidor ORDER BY DBMS\_RANDOM.VALUE) WHERE ROWNUM = 1;

    update envio set repartidor\_numero = rep\_id where numero\_envio = num\_env;

    if SQL%ROWCOUNT = 0 then

        dbms\_output.put\_line('No se encontró el envío, no se asignó repartidor');

    else

        dbms\_output.put\_line('Se asignó el repartidor ' || rep\_id ||' al envio número ' ||num\_env);

    end if;

END;

select \*from user\_procedures where object\_name = ‘ASIGNA\_REP’;

7. Conclusiones

Mediante el desarrollo y creación de la base de datos para el comercio electrónico (e-commerce) se cumplieron los objetivos propuestos, ya que la base de datos representa una mejora en los procesos, el control y la gestión para el comercio. Así mismo, se generó una estrategia en cuanto a los consumidores y competidores, ya que, con ésta, se le da un valor agregado, lo que hace que la empresa se dé a conocer en el ambiente electrónico. La base de datos, generó una reducción en los costos de operación, además ofrece comodidad en la adquisición del bien o producto y una mejor experiencia de usuario.

De esta manera se evita la pérdida de tiempo y costos significativos.   
Gracias a las características otorgadas se pudo desarrollar una solución conveniente además de adaptar las reglas del negocio del entorno físico al virtual, así como realizar la implementación en un modelo de datos de entidad relación y mediante él, construir un modelo de relacional con las características requeridas, esto para finalmente construir la base de datos utilizando las diferentes estructuras de SQL apropiadas para el proyecto.

A lo largo del proyecto se encontraron algunas dificultades como fue la integración de las reglas del negocio dentro del entorno de SQL ya que, eran dependientes del manejador de base de datos utilizado, por lo cual se debieron adaptar mediante el uso de estructuras pertenecientes al lenguaje PLSQL con el objetivo de hacerlas funcionales y dinámicas.

Finalmente, se puede concluir que la realización de la base de datos para el sistema de comercio electrónico, sirvió para mejorar la gestión de éste, además, se pudo adaptar el sistema que se tenía en el entorno físico a uno virtual, por lo que el comercio, ahora puede extender sus oportunidades de negocio, su alcance, clientes y ganancias; así mismo, ahora puede tener un registro de las operaciones realizadas y de esta manera, mejorar la gestión de la información y realizar un análisis que detalle la situación del comercio de forma más precisa y eficiente que en el esquema físico.