***Software de inventario y facturación para la empresa Tornillitos.***

[Carlos Mario Mejia Velasquez](mailto:cmejia6@udi.edu.co)

[Juan Diego Marquez Pineda](mailto:jmarquez4@udi.edu.co)

***Julián Daniel Macías Guerrero***

Proyecto Integrador.

Alexandra Soraya Beltran

Crisostomo Alberto Barajas

***UDI***

***Universidad de investigación y desarrollo***

# ***Introducción:***

En el centro de la Ciudad de Bucaramanga está ubicada una pequeña ferretería llamada Tornillitos, el dueño de la empresa Juan, empezó con ella hace unos años, el trabajaba solo y con algún familiar al que él le pagaba por ayudarle, y en el último año y medio dejo de ser una empresa familiar por varios factores que hicieron que el sector donde se ubica Tornillitos sea un lugar donde se necesite una ferretería, pues se están haciendo tres construcciones alrededor de esta, un edificio de apartamentos, un edificio de oficinas, y una remodelación de otro edificio existente, esto ha generado que Juan necesite de empleados y que las ventas suben exponencialmente, ahora necesita de un sistema de seguridad con cámaras, además de un sistema de organización de los empleados y la ferretería para que pueda atender a varios clientes al tiempo, actualmente tiene 5 empleados, pero son seis vendedores por que el aun atiende en el local, tiene 5 proveedores de los diferentes de productos.

Los productos que se venden en la ferretería son Martillos, Destornilladores, Clavos, Taladros,Pintura, Abrasivos,Tornillos, Tuercas,Pegamento para Madera.Alicates, Cables de corriente, y tuberías PVC;Estos en sus diferentes presentaciones y algunos con diferentes tamaños

Juan viendo que su empresa estaba creciendo y que en el ultimo año y medio casi triplico las ventas, por esto esta buscando nuevos metodos y tecnologias que puedan renovar, mejorar, e impulsar de buena manera la Ferreteria Tornillitos, la gestion que tiene con los empleados es la correcta, y ya que cada uno se encarga de una sección o varias, entonces en el mostrador los clientes van a la sección que necesitan piden sus productos, y luego de eso los vendedores hacen un vale con los productos que fueron entregados, luego de eso los clientes van a la caja y dan el vale y muestran los productos, la caja, es una caja registradora normal, y por el tedioso trabajo que se hace al poner un codigo a cada producto, sea de barras o tener algunos codigos en caja, en ese tipo de cajas registradora, el camino facil y más comodo es solo ingresar en el sistema de caja los ingresos de dinero y la salida de este, Juan ha pensado que personalizar la empresa a las comodidades y necesidades puede mejorar como se trabaja, por lo que ha pensado en que un sistema de caja espcificamente hecho para su empresa seria lo ideal, donde esto abarque todos los productos y sus precios, ademas de con la logica de caja que sigue su negocio.

En este documento haremos toda la planeación además de ir dando avances del desarrollo de manera paulatina y en orden.

# ***2) Descripción del problema.***

#### 2.1)¿Qué problemas presenta la ferretería Tornillitos.

La ferretería Tornillitos ha presentado un alza en el flujo de pedidos y cliente considerable , la infraestructura de la ferreteria no esta preparada ni puede responder ante la duplicación de pedidos y demanda de la ferreteria,el sistema de gestion manual de caja es demasiado lento y poco organizado para poder responder de manera eficiente a la clientela, tiempos de espera demasiado largas, descuadres en caja, perdida de facturas, y un conocimiento poco profundo de como se mueven las facturas y ventas dentro de la ferreteria, en si la gestion de facturas y caja de manera manual se ha mostrado ineficiente, incompleto e incompetente para poder satisfacer las necesidades de la ferreteria.

#### 2.2)¿Cómo se puede solucionar el problema?

La implementación del software será la facturación de pedidos, y el registro de ventas de manera profesional, deberá manejarse todos los aspectos de los pedidos y que estén claros para el cliente, tales como la cantidad, tipo, número de factura, valor, información del cliente, además de esto en el registro de ventas que manejaremos los productos dependiendo del tipo de producto ya que pueden tener varios tamaños y presentaciones, se añadirán después de su nombre signos para identificar tipo y tamaño, el registro de pedidos tendrá que ser fácil de manejar para los administradores y los clientes podrán ver sus facturas.

Se tendrán que hacer reportes a la administración, para saber el estado de las ventas.

# 

# 

# 

# ***Objetivos.***

## ***Generales.***

Se debe planificar, desarrollar e implementar un software para optimizar el sistema de caja,reportes de ventas y facturación de pedidos, así como establecer un registro eficiente de ventas para la ferretería Tornillos.

## ***Específicos.***

- Analizar el sistema de organización y el funcionamiento interno que tiene la ferretería Tornillitos, como se comporta y cómo se relaciona, para representar estos en los diagramas UML.

- Utilizar Diagramas UML para poder visualizar la estructura óptima del código y diseñar una solución eficiente que satisfaga las necesidades identificadas.

- Codificar el software conforme a los requerimientos y los diagramas UML elaborados previamente ,realizando las pruebas necesarias(aceptación,unidad y componentes) para garantizar su correcto funcionamiento.

- Implementar el software en la infraestructura de la ferretería Tornillitos, proporcionando la documentación y capacitación necesaria para su uso efectivo por parte del personal.

# 

# 

# 

# ***Justificación.***

La implementación del software en la pyme Tornillitos es un gran recurso además de necesario, puesto que además de que se está retrasando los procesos de la pyme, el flujo de personas y de caja, se retrasan las posibles ventas, la optimización podría triplicar los procesos de factura y de pedidos, esto hace que tal vez los 25 pedidos diarios que se atienden en la Ferretería, podrían triplicarse, no optimizar los sistemas necesarios para que un local, negocio o empresa pueda atender la demanda de sus productos que produce pérdidas de dinero, recursos, tiempo, clientes, oportunidades e incluso un mal nombre para la empresa pues los clientes se hartaran de la falta de organización y falta de profesionalismo en la empresa.

El software que implementamos tiene la intención de mejorar organización económica y monetaria de la empresa, los reportes de ventas constantes mejoran el control de las ventas y de producción que se puede tener, si se conocen los procesos a la perfección de la empresa matemáticamente con exactitud se pueden optimizar los métodos de ventas ya que en algunas empresas pyme ni siquiera los dueños conocen con exactitud todas sus ventas y gastos.

Este proyecto mejorará no solo de forma organizacional y productiva, también hará que la calidad de procesos, como la atención al cliente, ventas, calidad sistémica en general, potenciando así su ya natural ascenso para mejorar la empresa.

***Propuesta plan de proyecto***

Empezaremos planteando el problema de la empresa Tornillitos, se harán los diagramas para entender y graficar el funcionamiento de la empresa, se harán los diagramas entidad-relación y relacional, para entender cómo va a funcionar la base de datos, se crearán los diseños de interfaces, se codifica, harán pruebas para el funcionamiento y se implementará el software en la pyme.

utilizaremos Algunas herramientas que nos han enseñado en clases, para la base de datos se utilizará para las vistas y la interfaz del programa JAVA, utilizaremos oracle express 10g para la base de datos, y para el diseño de las interfaces utilizaremos canva, la programación se hará en su mayoría con java, y para los modelamientos de la base de datos utilizaremos programas como lucidchart, o draw.io, especializados en crear diagramas. se presentará en una carpeta en google drive con todo el contenido que se necesite.

Calendario proyecto.

20-27 de Febrero hicimos el planteamiento del problema y se empezaran a hacer los primeros bocetos de diagramas UML.

28 de febrero -6 de marzo realizamos los diagramas de UML, de casos de uso y de secuencia.

6 marzo - 13 de Marzo Hicimos los diagramas Entidad-Relación, Relacional, y Diccionario de datos.

13-16 Marzo Realizamos en canva los diseños de las interfaces.

***Análisis de requerimientos de software.***

#### Requerimientos funcionales.

- Proceso de Facturación: Asegurar el correcto funcionamiento del proceso de facturación,permitiendo la creación de facturas para las transacciones realizadas recopilando información como lo es: la cédula del cliente,la cédula del vendedor,fecha un subtotal y el IVA.

- Generación de Reportes de Ventas: Desarrollar la funcionalidad para generar reportes de ventas que proporcionen una visión completa de las actividades comerciales, presentando información detallada de las ventas como la id de la factura,en que fecha se realizó, qué costo total tuvo y a que persona se le atribuyó esa factura.

- Registro de Ventas: Garantizar el registro adecuado de todas las ventas realizadas para mantener un historial completo y preciso,la información de la venta se almacenará en la base de datos para una futura consulta que se requiera.

- Operaciones CRUD de ventas: Establecer operaciones CRUD(Crear,Leer,Actualizar,Eliminar) funcionales para facturas y reportes

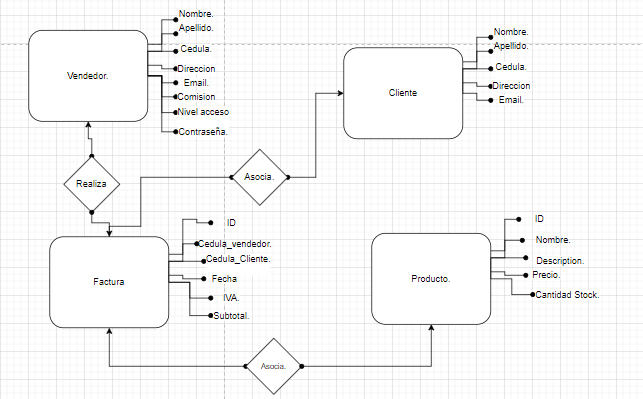
#### Requerimientos no funcionales.

- Login distintivo: Se deberá realizar un login distintivo para el vendedor.

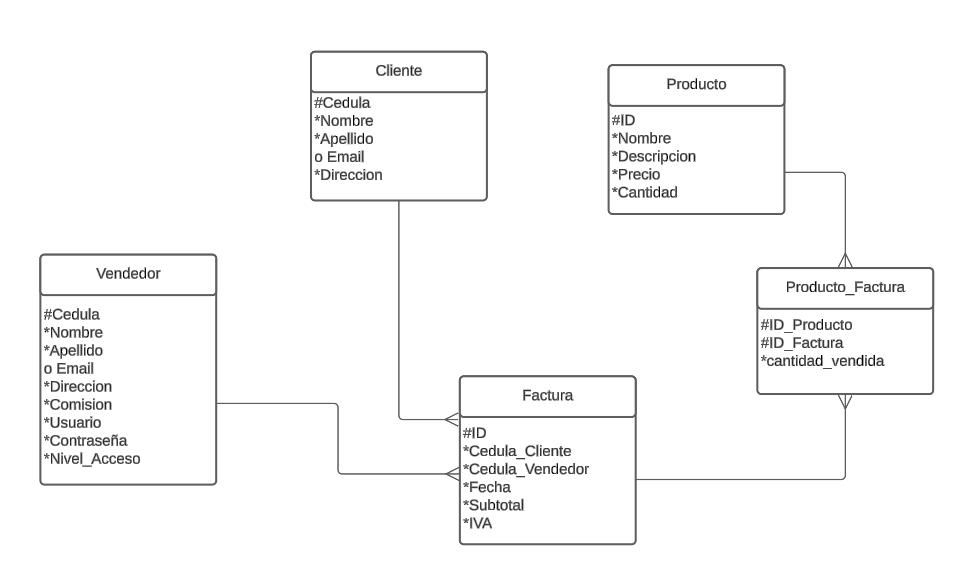
- Diseño Visual: Diseñar las vistas del software con una paleta de colores que incluya tonos amarillos y cafes.

- Lista de Productos Etiquetada: Mantener una lista etiquetada de los productos disponibles en la ferretería para su venta,facilitando la selección y compra por parte de los clientes.

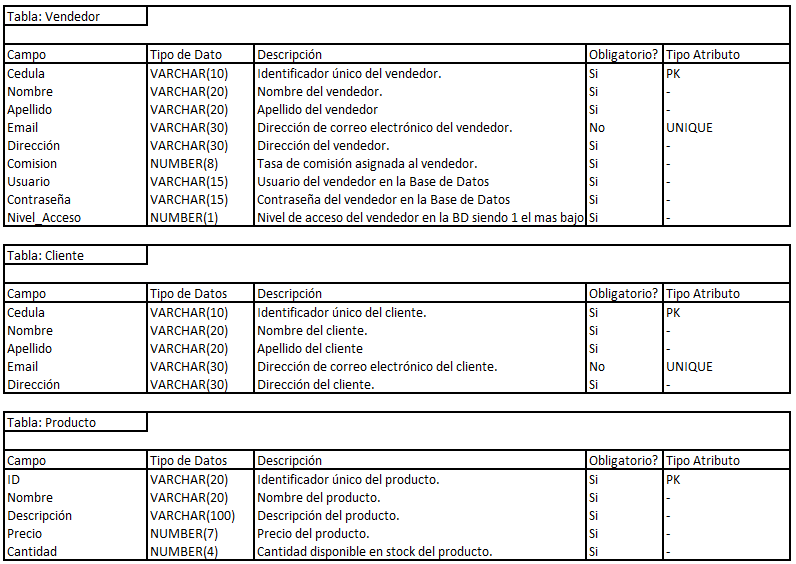
***Modelo Entidad-Relación***

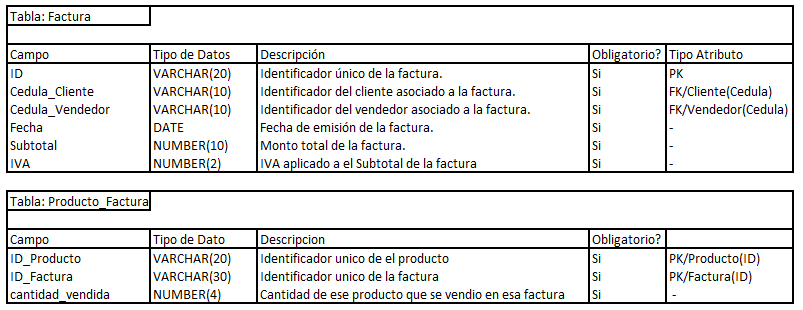


***Modelo Relacional:***

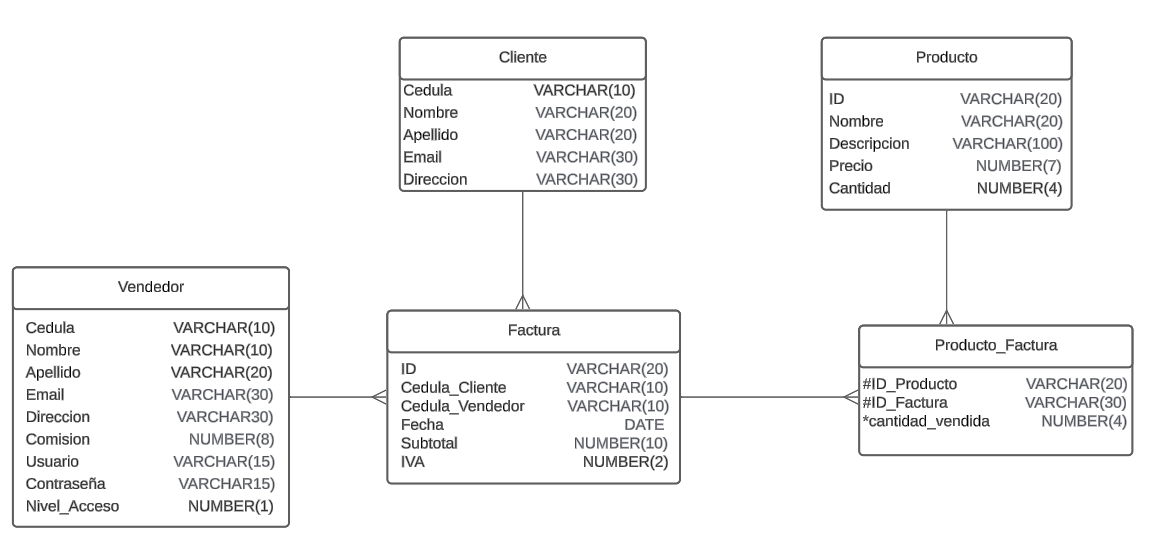


***Diccionario de datos.***





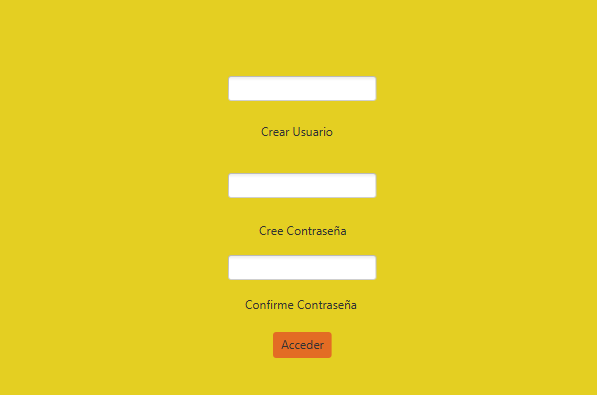
***TablasSQL***



***Diseño de Interfaz:***

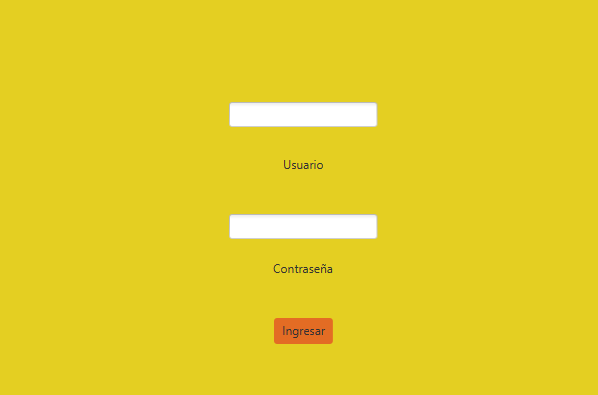
Página de ingreso:

En el Scene Builder tomamos un Pane de menú de containers al que le colocaremos el color #E4CF22 con la opcion de color en el style, creamos un logo referente al nombre de la ferretería,luego añadimos un label con la palabra “bienvenido” además de dos botones sacados del menú de controladores, con la opción de acceder y crear usuario.

Crear Usuario:

Cuando le damos click al botón de crear usuario se nos abre esta ventana en la cual cogemos 3 label al que les pondremos crear usuario, cree contraseña y confirme contraseña, y arriba del label de crear usuario ponemos un textfield para que lo ingresemos, y en el de crear contraseña y confirmar ponemos dos passwordfield.

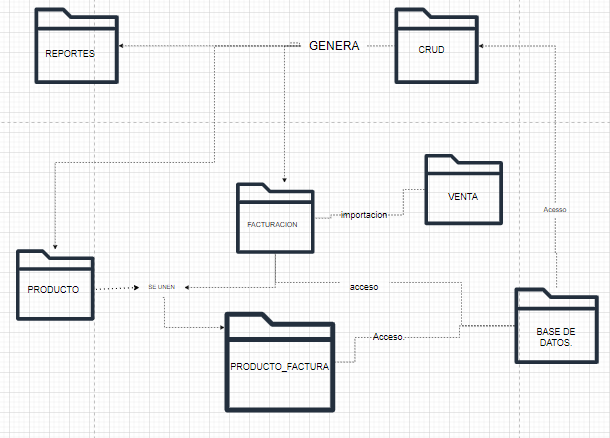
Acceder:



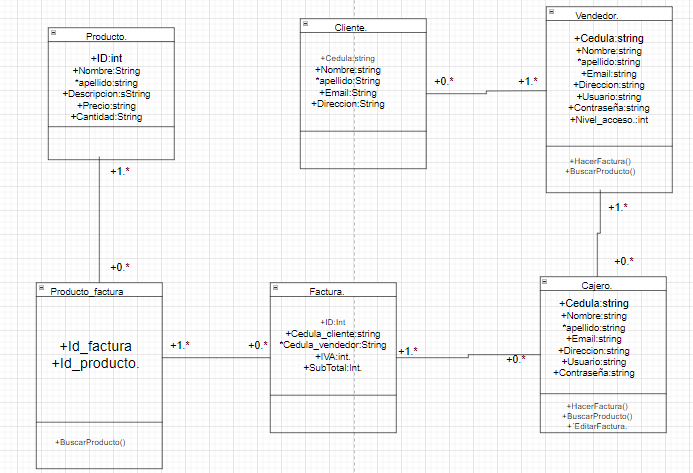
por último cuando le damos click al botón de acceder se nos abre esta ventana en la cual utilizamos dos label que son el de usuario y contraseña y para que se pudieran ingresar utilizamos un textfield y un passwordfield.

DIAGRAMAS UML.

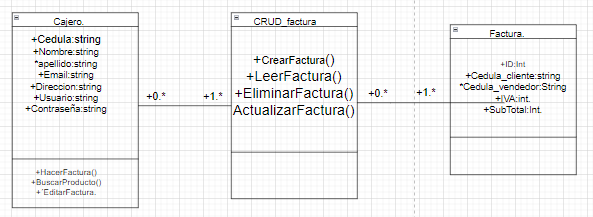
#### -Diagrama de paquetes.



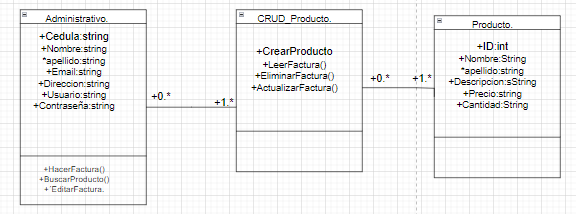
#### -Diagram de clases de proceso de venta.



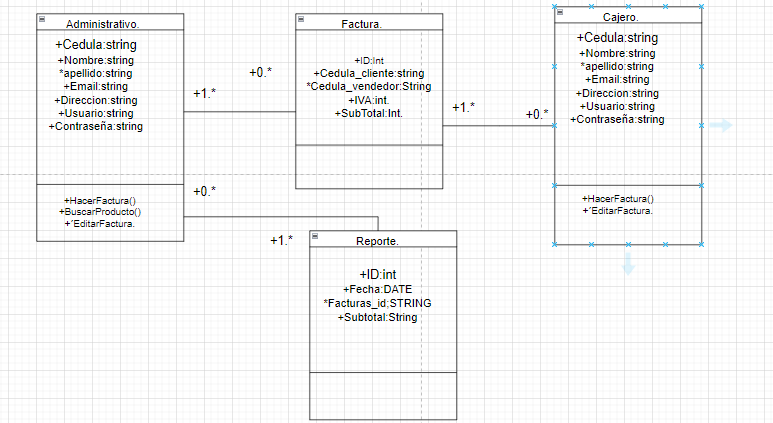
#### -Diagrama de crud factura.



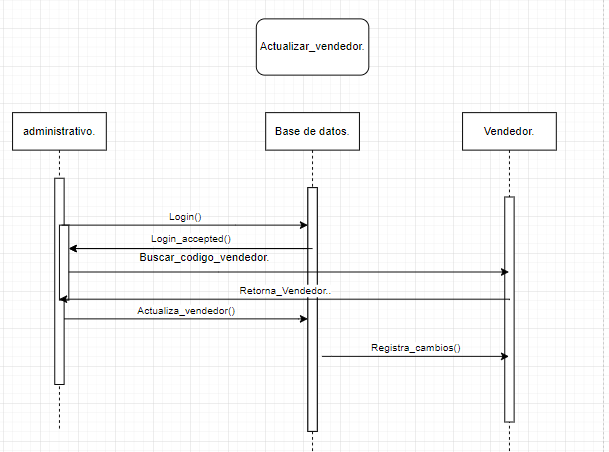
#### -Diagrama de clases crud de producto.

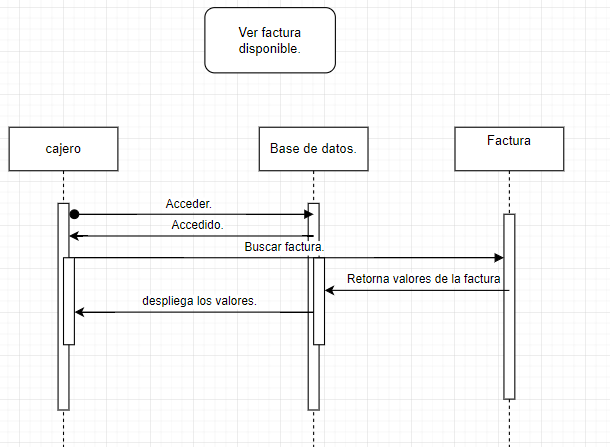


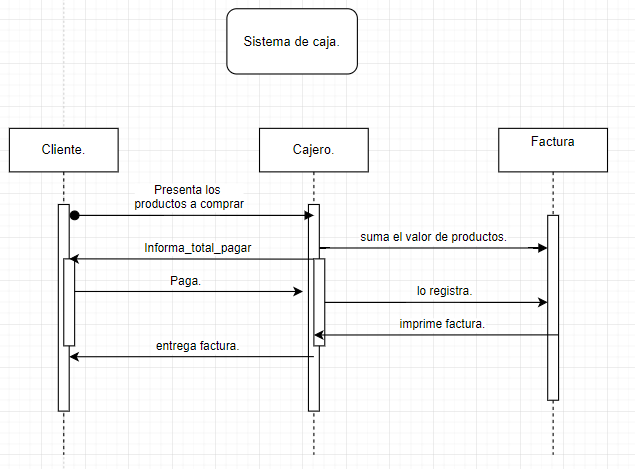
#### -Diagrama de CRUD de Reportes de ventas.

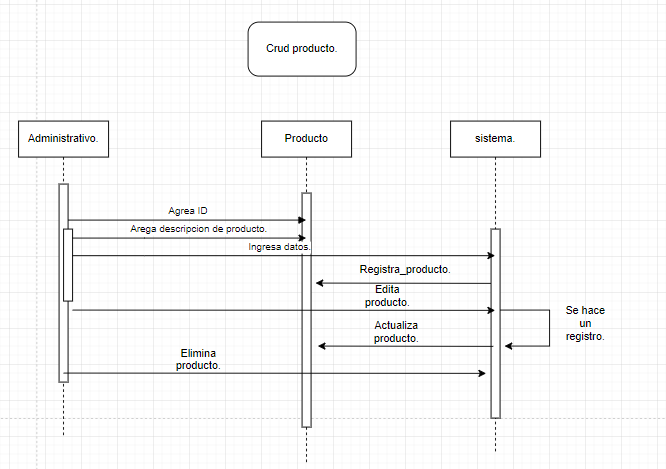


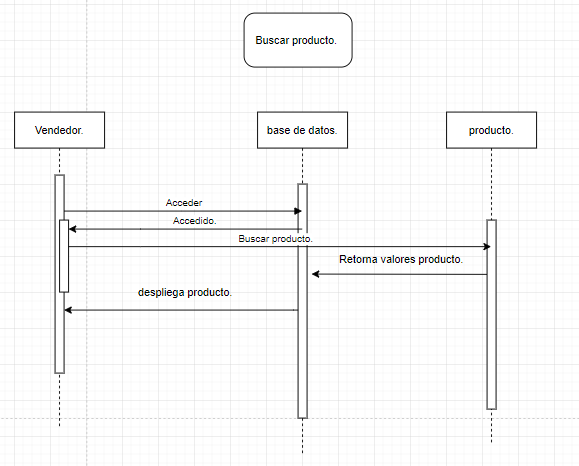
#### Diagramas de secuencia.

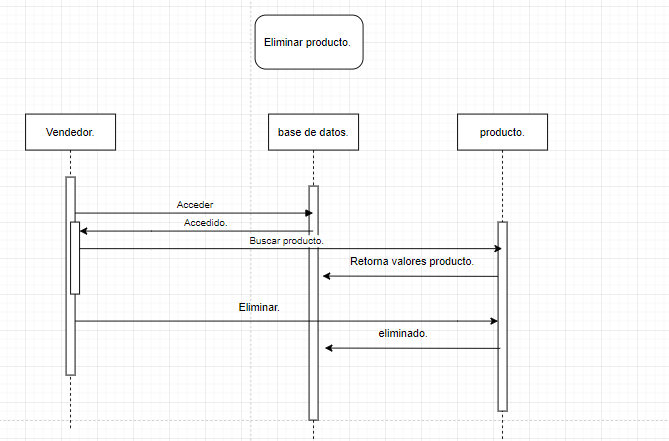


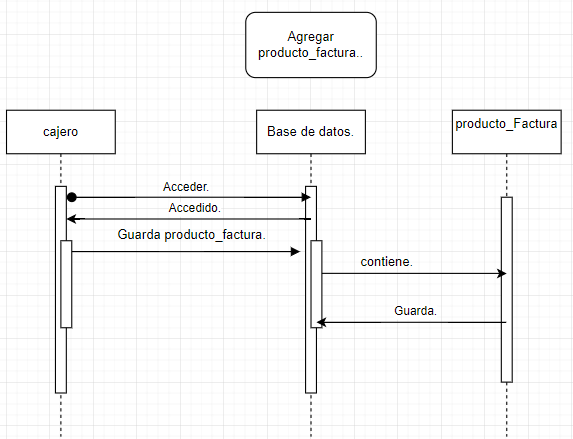


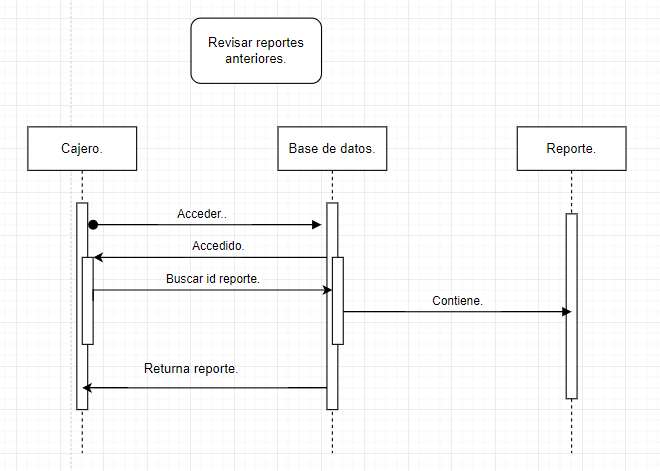


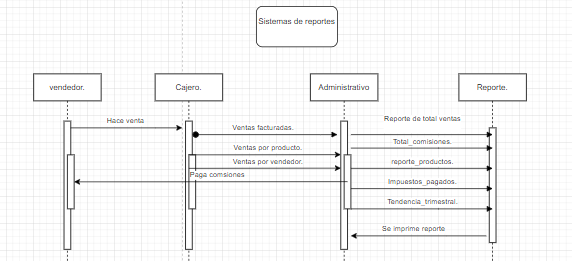


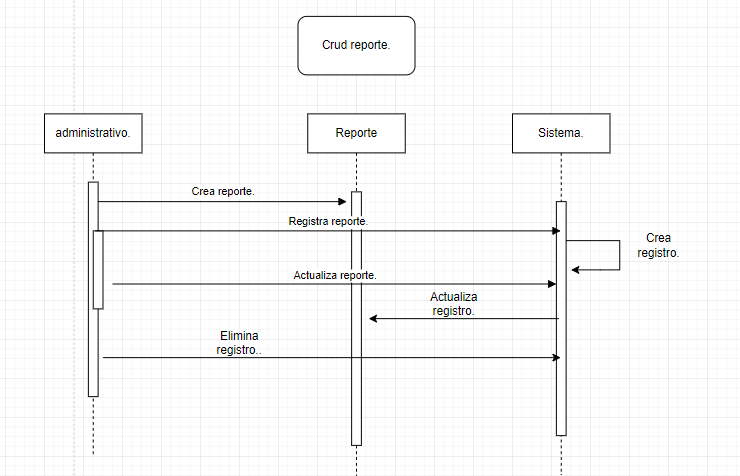


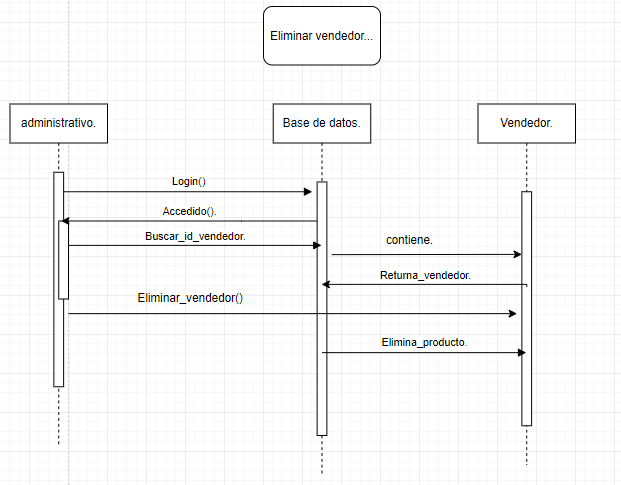


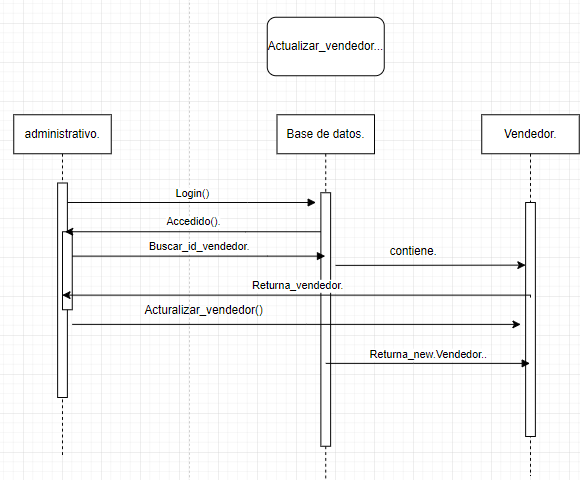


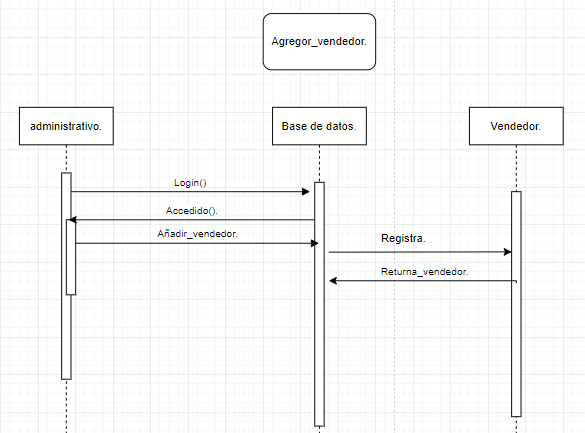




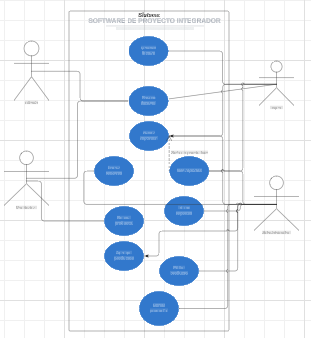




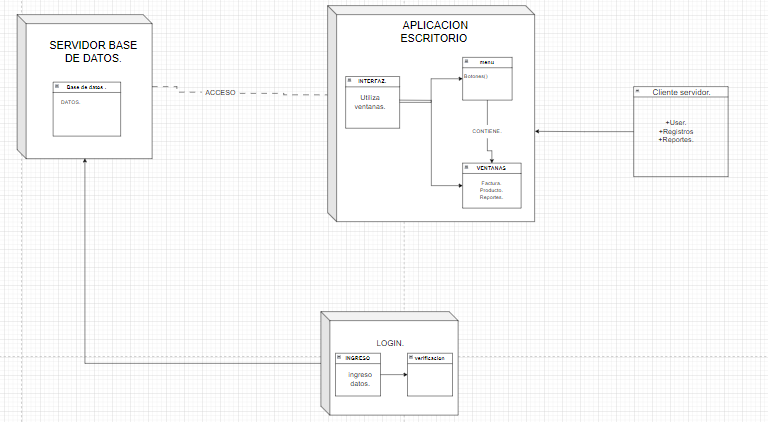




Diagramas de caso de uso.



#### Diagrama de despliegue.



Creación de tablas SQL

Usuario: Proyecto.

contraseña:Proyecto.

Tabla Cliente:

SQL> create table Cliente(

2 cedula varchar(10) constraint cli\_ced\_pk primary key,

3 nombre varchar(20) constraint cli\_nom\_nn not null,

4 apellido varchar(20) constraint cli\_ape\_nn not null,

5 email varchar(30) UNIQUE,

6 direccion varchar(30) constraint cli\_dir\_nn not null);

Tabla Vendedor:

Create table Vendedor(

2 Cedula varchar(10) constraint ven\_ced\_pk primary key,

3 nombre varchar(20) constraint ven\_nom\_nn not null,

4 apellido varchar(20) constraint ven\_ape\_nn not null,

5 email varchar(30) UNIQUE,

6 direccion varchar(30) constraint ven\_dir\_nn not null,

7 comision number(8) constraint ven\_com\_nn not null,

8 usuario varchar(15) constraint ven\_usu\_nn not null,

9 contrasena varchar(15) constraint ven\_con\_nn not null,

10 nivel\_acceso number(1) constraint ven\_niv\_acc\_nn not null);

Tabla Producto:

create table producto(

2 id varchar(20) constraint pro\_id\_pk primary key,

3 nombre varchar(20) constraint pro\_nom\_nn not null,

4 descripcion varchar(100) constraint pro\_des\_nn not null,

5 precio number(7) constraint pro\_pre\_nn not null,

6 cantidad number(4) constraint pro\_can\_nn not null);

Tabla Factura:

create table Factura(

2 id varchar(30) constraint fac\_id\_pk primary key,

3 Cedula\_cliente varchar(10) constraint fac\_ced\_cli\_nn not null,

4 constraint fac\_ced\_cli\_fk foreign key(Cedula\_cliente)

5 references Cliente(cedula),

6 Cedula\_vendedor varchar(10) constraint fac\_ced\_ven\_nn not null,

7 constraint fac\_ced\_ven\_fk foreign key(Cedula\_vendedor)

8 references Vendedor(cedula),

9 fecha date constraint fac\_fec\_nn not null,

10 subtotal number(10) constraint fac\_sub\_nn not null,

11 IVA number(2) constraint fac\_iva\_nn not null);

Tabla Producto\_Factura:

create table Producto\_factura(

id\_producto varchar(20) constraint pro\_fac\_id\_pro\_fk references producto(id),

id\_factura varchar(30) constraint pro\_fac\_id\_fac\_fk references factura(id),

cantidad\_vendida number(4) constraint pro\_fac\_can\_ven\_nn not null,

constraint pro\_fac\_idfac\_idpro\_pk PRIMARY KEY (id\_producto,id\_factura));

Niveles de Acceso:

Nivel 1(Vendedores): En este nivel los usuarios podrán insertar datos dentro de las tablas facturas, clientes y producto\_factura pero no podrán modificar la estructura de los datos.Por ejemplo el vendedor Juan puede hacer una factura pero primero necesita los datos del cliente así que él los ingresa y posteriormente hace la factura pero no puede cambiar la información de un cliente al menos sin un permiso especial.

Nivel 2(Gerente): En este nivel los usuarios podrán insertar,cambiar o modificar los datos de todas las tablas pero no su estructura.Ejemplo: Jorge(gerente) puede cambiar la información de los vendedores o de los clientes pero no puede cambiar la estructura(tipos de datos,constraints,etc)

Nivel 3(Administrador de la base de datos): En este nivel los usuarios podrán cambiar todo en la base de datos en esto incluye(datos,estructura y composición de los datos).

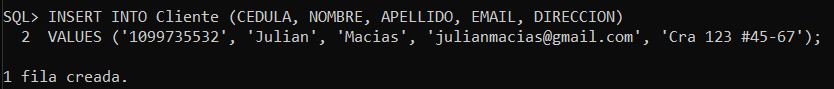
Verificación de requerimientos

Proceso de Creación de una Factura:

Siendo la vendedora Estefania con número de cédula 129878.

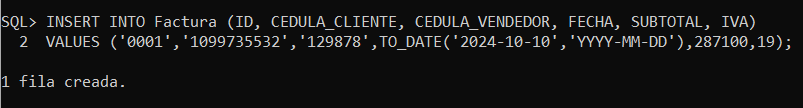
Paso 1:

se ingresa la información de un cliente:



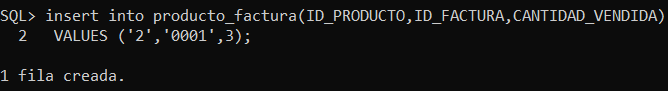
Paso 2:

Se ingresa la información a la factura teniendo en cuenta la cc de la vendedora(129878), la cc de el cliente(1099735532),dentro de el subtotal va incluido el IVA, en este caso el señor Julian va comprar 3 destornilladores cada uno por un valor de $90.000 por lo que serían 270.000 y a eso le agregamos el IVA de un 19% entonces quedaría en un valor de $287.100



Paso 3:

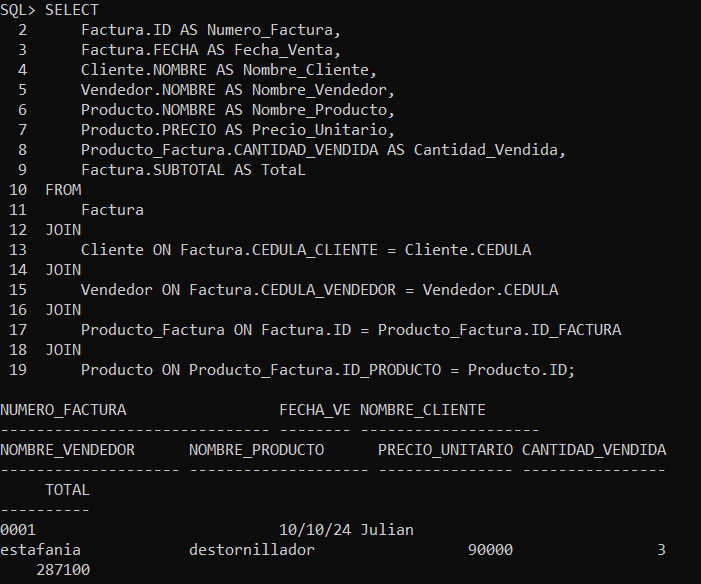
Ingresar la información anterior en la tabla producto\_factura para un posterior reporte de las facturas.



Para este caso Julian solo compro un producto si hubiera comprado 2 o 3 productos habría que hacer el procedimiento anterior 2 o 3 veces dependiendo de qué productos compró.

Generación de reporte de Facturas y Registro de Ventas:

El requisito de mantener un historial completo y preciso de todas las transacciones comerciales, tal como se establece en el requerimiento de 'Registro de Ventas', se está cumpliendo a través de la funcionalidad existente de generación de reporte de factura en la base de datos. Esta funcionalidad permite acceder y presentar de manera estructurada la información detallada de cada venta, incluyendo la ID de la factura, la fecha de la venta, el costo total y la persona a la que se atribuyó esa factura. Estos datos están previamente almacenados en la base de datos como parte del proceso de registro de ventas, asegurando así un historial completo y preciso de todas las transacciones comerciales realizadas.



select

factura.id AS Numero\_Factura,

factura.fecha AS Fecha\_Venta,

cliente.nombre AS Nombre\_Cliente,

vendedor.nombre AS Nombre\_Vendedor,

producto.nombre AS Nombre\_Producto,

producto.precio AS Precio\_Unitario,

producto\_factura.cantidad\_vendida AS Cantidad\_Vendida,

FROM

Factura

JOIN

cliente ON factura.cedula\_cliente = cliente.cedula

JOIN

vendedor ON factura.cedula\_vendedor = vendedor.cedula

JOIN

producto\_factura ON factura.id = producto\_factura.id\_factura

JOIN

producto ON producto\_factura.id\_producto = producto.id;

Operaciones CRUD de las ventas

Llenado de datos.

<https://drive.google.com/file/d/1z9oth_nEQEAYW0pIwgZT_hVNDcCZWpi6/view?usp=sharing>

Cambios que se han hecho:

\* En la tabla Vendedor se modificó el tipo de dato de la columna Comision a number(8) debido a que oracle no permite el tipo de dato INT.

\* En la tabla Vendedor se modificó el tipo de dato de la columna Nivel\_acceso a number(1) debido a que oracle no permite el tipo de dato INT.

\* En la tabla Producto se modificó el tipo dato de la columna Precio a number(7) debido a que oracle no recibe el tipo de dato INT.

\* En la tabla Producto se modificó el tipo dato de la columna Cantidad a number(4) debido a que oracle no recibe el tipo de dato INT.

\* En la tabla Factura se modificó el tipo de dato de la columna Subtotal a number(10) debido a que oracle no recibe el tipo de dato DECIMAL.

\* En la tabla Factura se modificó el tipo de dato de la columna IVA a number(2) debido a que oracle no recibe el tipo de dato DECIMAL.

\* En el diccionario de datos se modificaron los tipos de datos de los anteriores puntos.

\* Se mejoró la redacción de los Objetivos

\* Se añadió una descripción a los requisitos para poder evaluar el modelo de datos relacional.

\* Se añadió tanto al modelo de datos como al diccionario de datos el campo de cantidad\_vendida y de tipo number(4) para poder registrar la cantidad de un producto de una factura.

\* Se cambió la vista de las tablas SQL para mejorar la legibilidad de las tablas y que tipo de datos son.

\* Se añadió una sección de procedimientos para verificar el cumplimiento de los requisitos.

\*a la base de datos se le quitaron las restricciones unique en los campos de id\_factura,id\_producto a la tabla producto\_factura.

\*se ha corregido el diagrama entidad relacion con los simbolos correspondientes.