



UNIVERSIDAD RAFAEL LANDÍVAR

FACULTAD DE INGENIERÍA

ANÁLISIS Y DISEÑO II

PROYECTO MEVECON

JORGE STEVENS IZÁS ROJAS – 1508717

JOSE PABLO SANCHEZ ESTRADA - 1577871

GUILLERMO DANILO HERRERA CASTILLO - 1509916

QUETZALTENANGO JULIO DEL 2020

Proyecto Análisis y Diseño II

Información del negocio:

- **Nombre de la empresa:** Distribuidora Mevecon
- **Ubicación:** Diagonal 2, 27-49 zona 3 Quetzaltenango
- **Descripción:** Negocio dedicado a la venta de alimento para animales como perro, gatos, gallinas, aves y también pastillas, medicinas, entre otros productos por mayor y menor ubicado en Quetzaltenango.

Idea para el Sistema:

- Se escogió la Distribuidora Mevecon, ya que su modelo de negocios era todo a mano, desde el control de facturas hasta el control del inventario, ya que es una distribuidora que vende por menor y mayor se vio la necesidad de proponer el desarrollo de un software al gerente de la distribuidora esto con la finalidad de que logran llevar mejor control de las ventas, inventario, compras, control de producto entre otros para mejorar las operaciones de la empresa.

Módulos a trabajar

A continuación, se listan los módulos que se trabajarán en el proyecto. Cada columna es un módulo, en la parte superior está el nombre de dichos módulos y por debajo sus funcionalidades

Clientes	Inventario	Compras	Ventas	Proveedores	Usuarios
Registrar clientes nuevos	Listar inventario existente	Registrar compras	Registrar ventas	Registrar proveedores	Registrar usuario
Modificar clientes	Reservar inventario para compra posterior	Listar historial de compras	Aplicar descuentos	Listar proveedores	Asignar permisos
Listar clientes	Emitir alerta de bajo inventario	Modificar compra	Listar historial de ventas	Modificar proveedores	Modificar datos
	Mostrar descripciones de productos	Buscar factura	Anular ventas		
	Mostrar ubicaciones de productos	Actualizar inventario	Buscar factura		
			Modificar venta		
			Actualizar inventario		

Diagramas, Características y Requerimientos

Diagrama De Contexto

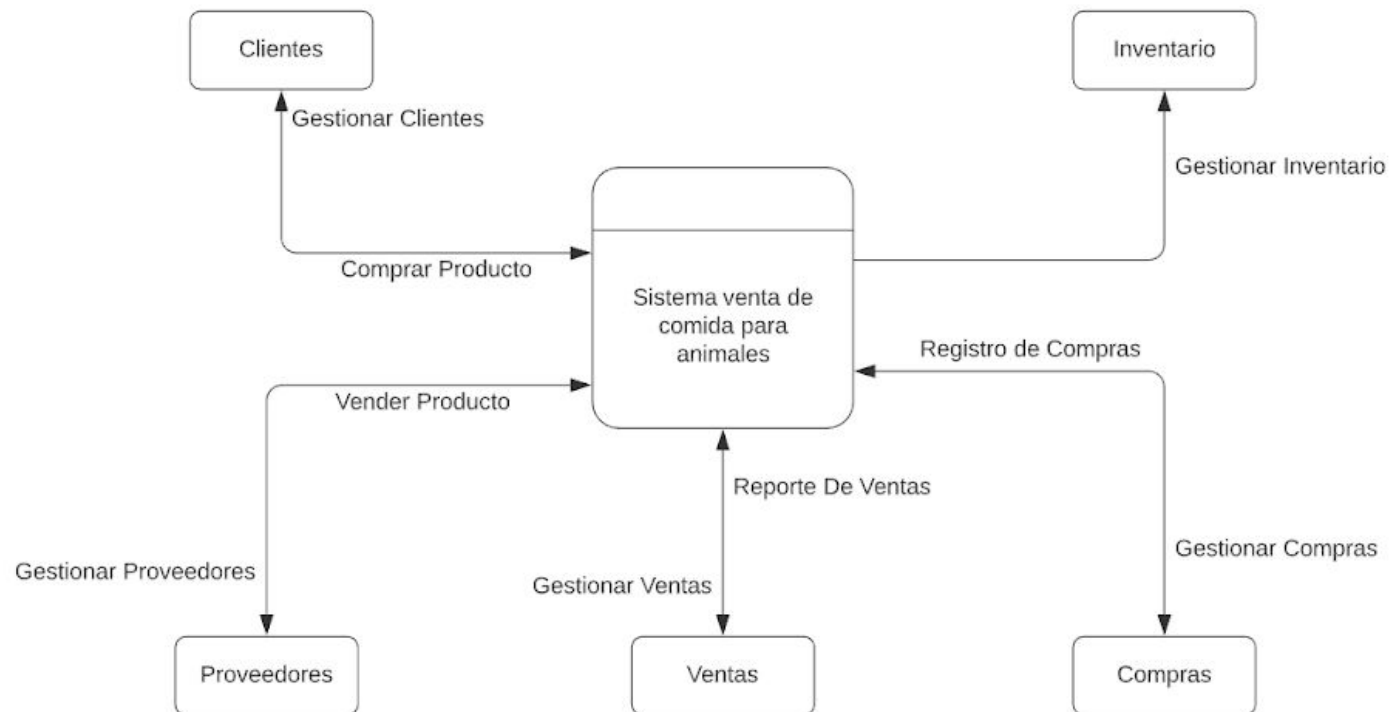
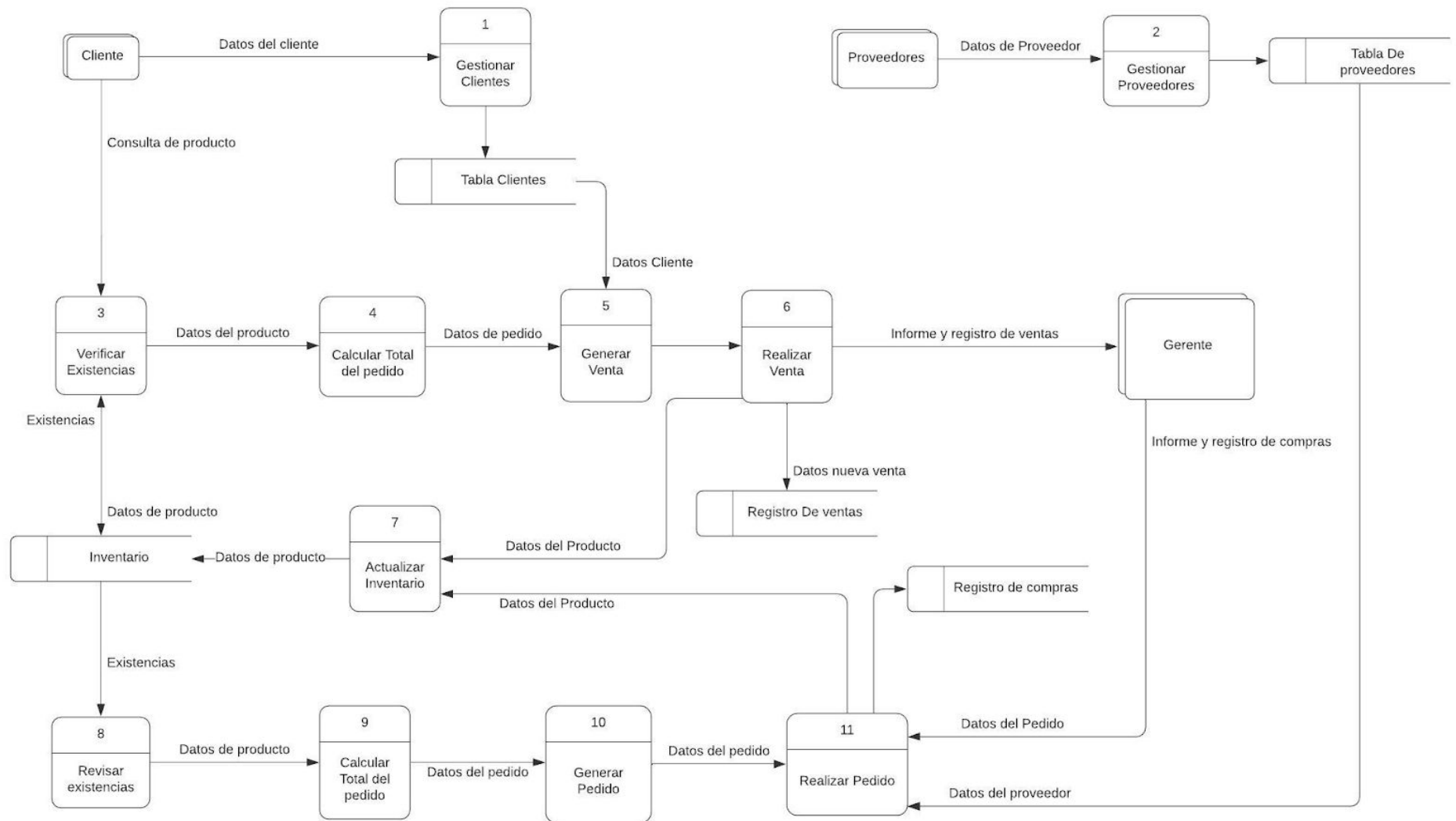


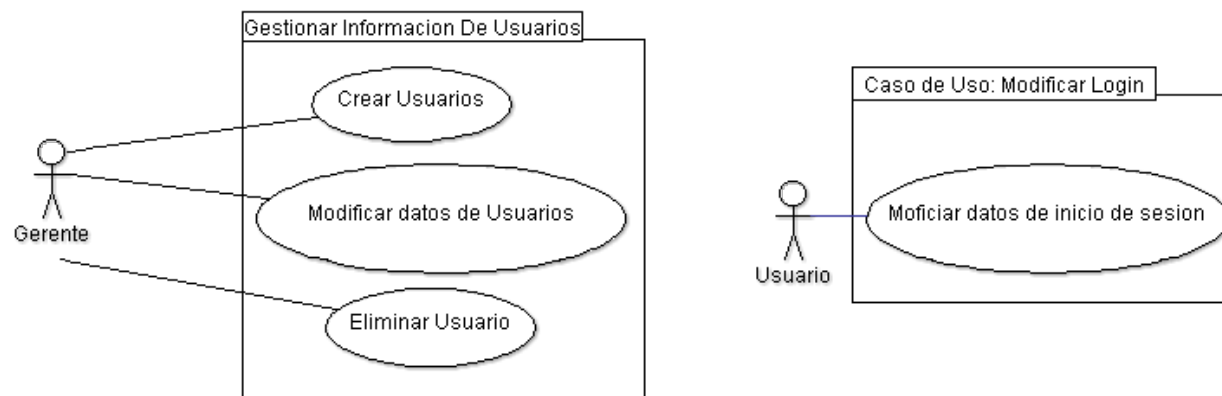
Diagrama 0



Diagramas de caso de uso

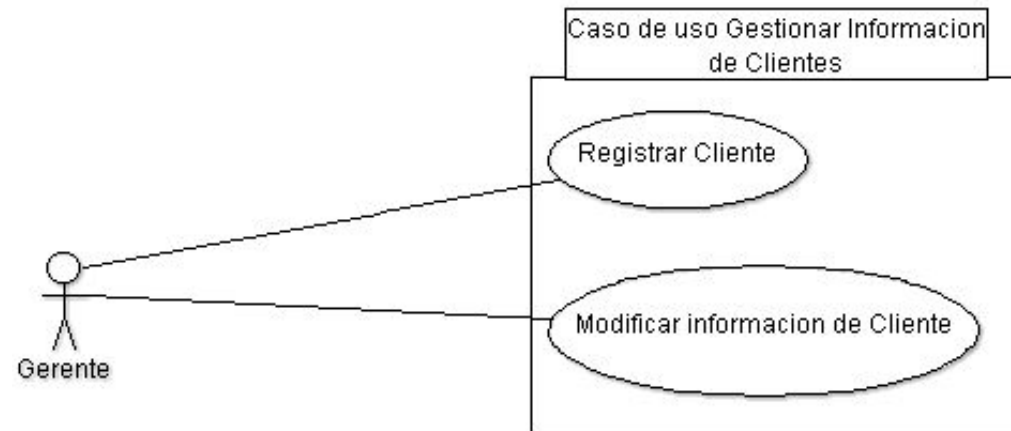
Gestionar Información de Usuarios

Características	Requerimientos
Crear Usuario	<ul style="list-style-type: none"> • Solo el gerente puede crear usuarios. • No pueden existir dos usuarios con el mismo ID, Nombre, Apellido, Correo, Teléfono • Cada usuario tendrá un ID diferente, de preferencia correlativo. • Los usuarios tendrán datos de acceso: Nombre, contraseña, permisos de acceso. • No se puede ingresar al sistema sin usuario. • Los nombres de usuario deberán usarse en las transacciones del sistema.
Modificar Usuario	<ul style="list-style-type: none"> • El Gerente puede modificar los permisos de acceso de los usuarios. • El gerente puede modificar los datos del usuario: Nombre, Apellido, Correo, Teléfono.
Modificar Datos de Inicio de Sesión	<ul style="list-style-type: none"> • El usuario puede cambiar su contraseña. • El usuario puede cambiar su "Nombre de usuario" para iniciar sesión
Eliminar Usuario	<ul style="list-style-type: none"> • Se deben eliminar los datos de acceso del usuario en la base de datos. • Habilitar el nombre de usuario después de eliminar el usuario.



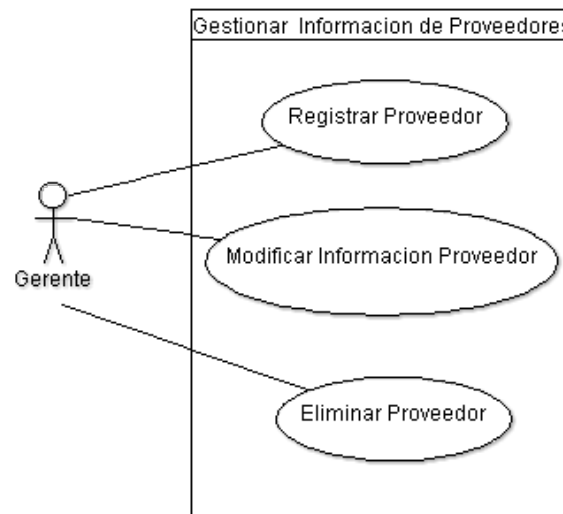
Gestionar Información de Clientes

Características	Requerimientos
Registrar Cliente	<ul style="list-style-type: none"> • Cada cliente tendrá un ID, Nombre, Apellido, Nit, Celular, Dirección. • El ID debe ser distinto para cada cliente. De preferencia correlativo • No pueden existir dos clientes con el mismo ID, Nombre, Apellido. • Debe haber un cliente "Consumidor Final" • Asignar Tipo de cliente: Frecuente/Normal • Asignar Nivel de clientes: Mayoristas/Minoristas
Modificar Cliente	<ul style="list-style-type: none"> • No se puede modificar el ID • Modificar Nombre, Apellido, Nit, Celular, Dirección • Modificar Tipo de cliente • Modificar Nivel de cliente • Modificar si un cliente esta Activo/Inactivo



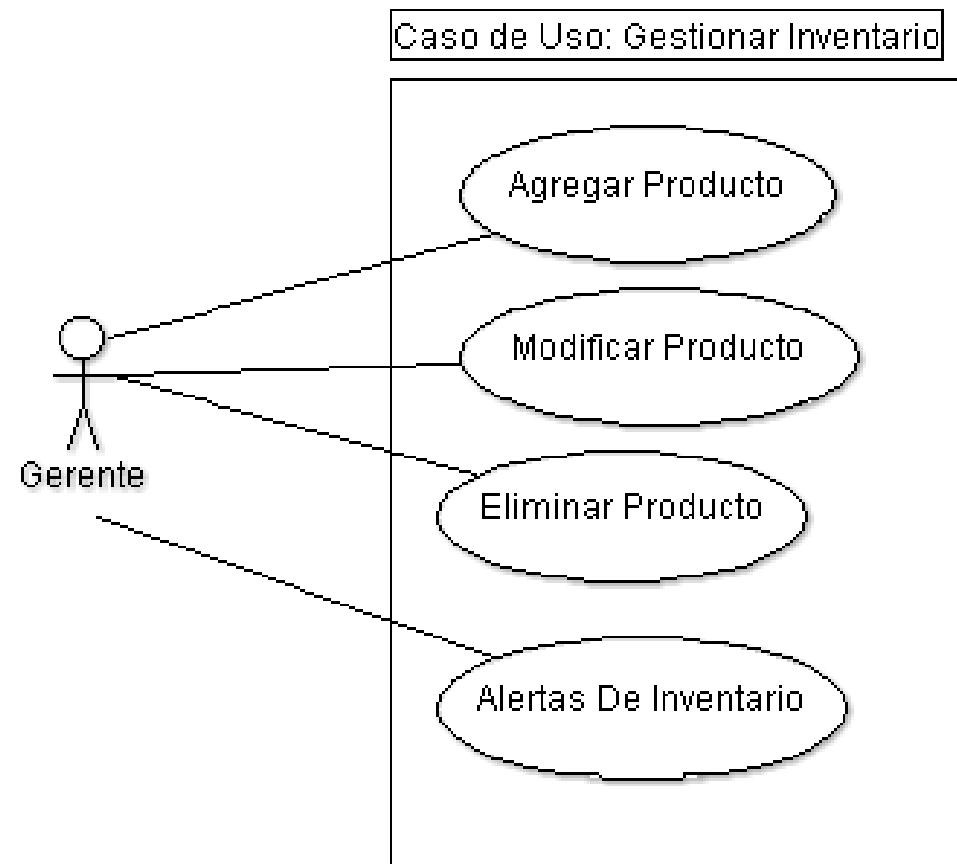
Gestionar información de proveedores.

Característica	Requerimiento
Registrar Proveedor	<ul style="list-style-type: none"> • Cada proveedor tendrá un ID, Nombre, Apellido, Celular, Dirección, Empresa • El ID debe ser distinto para cada proveedor. De preferencia correlativo • No pueden existir dos proveedores con el mismo ID, Nombre, Apellido • Definir los productos o marcas que provee
Modificar Información de proveedor	<ul style="list-style-type: none"> • No se puede modificar el ID • Modificar Nombre, Apellido, Celular, Dirección, Empresa. • Modificar los productos a su cargo
Eliminar Proveedor	<ul style="list-style-type: none"> • Eliminar datos del proveedor • Marcar como inactivo la compra de los productos a cargo del proveedor



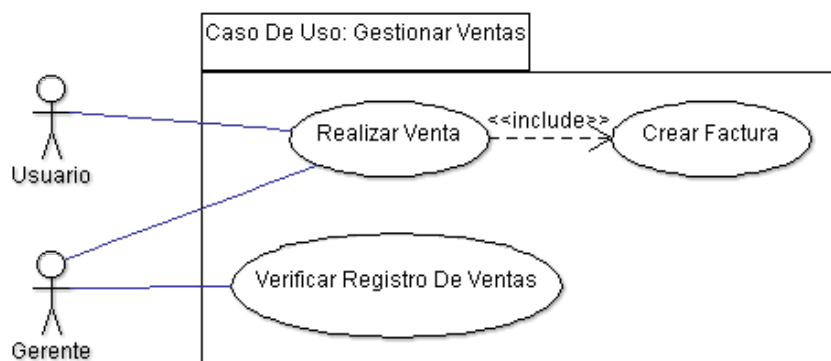
Gestionar Inventario

Características	Requerimientos
Agregar productos al inventario	<ul style="list-style-type: none"> • Cada Producto Tiene un ID único • Agregar información necesaria del producto: Nombre, Marca, Presentación, Categoría, Fecha de Vencimiento, Fecha de ingreso. • Agregar la cantidad de producto que se está ingresando • Actualizar el inventario cada vez que se agrega o edita un producto • Registrar el día que se ingresó el producto en la base de datos • Buscar productos y poder agregar filtros de búsqueda
Modificar información de productos	<ul style="list-style-type: none"> • No se puede modificar el ID • Es posible modificar Nombre, Marca, Presentación, Categoría, Fecha de Vencimiento, Fecha de ingreso.
Eliminar productos del inventario	<ul style="list-style-type: none"> • Registrar el día que se eliminó el producto en la base de datos • Definir si se va a eliminar una cantidad del producto o el producto por completo • Actualizar el inventario después de eliminar el producto
Alertas de inventario	<ul style="list-style-type: none"> • Mostrar listado de transacciones de inventario: fechas en que se agregó, eliminó, vendió, compró producto. • Alerta del producto que está próximo a vencer. La alerta deberá mostrarse con días de anticipación definidos por el gerente Mostrar una alerta diaria hasta el día de vencimiento. • Alerta de existencias mínimas. El gerente debe definir el mínimo de existencias que deben estar en el inventario Opción de adquirir existencias contactando al proveedor



Gestionar Ventas

Características	Requerimientos
Realizar Venta	<ul style="list-style-type: none"> • Guardar los datos de la venta (Usuario, cliente, productos, fecha, total) • Verificar si el cliente de la venta existe o es necesario el ingreso. • Se descuenta el producto que se vendió hasta que se termina la venta y se actualiza el inventario. • Opción de agregar un descuento en la venta. • Mostrar si el cliente es frecuente para realizar el descuento • Editar datos de la venta antes de finalizarla • Buscar productos por categoría o filtrados para agregarlos a la venta • Definir si la venta es en la tienda o a domicilio • Crear la factura correspondiente a la venta.
Verificar Registro De Ventas	<ul style="list-style-type: none"> • Buscar ventas realizadas por categorías o filtros (Fecha, cliente, producto) • Verificar los ingresos según las ventas que se realizaron • Informe de ventas semanal, mensual, anual.



Gestionar Compras

Características	Requerimientos
Realizar Compra	<ul style="list-style-type: none">• Guardar los datos de la venta (Proveedor, producto, fecha, total)• Actualizar los precios.• Actualizar el inventario• Verificar que el proveedor de la venta exista o ingresar un nuevo proveedor
Verificar Registro De Compras	<ul style="list-style-type: none">• Buscar compras realizadas por categorías o filtros (Fecha, proveedor, producto)• Informe de compras semanal, mensual, anual.

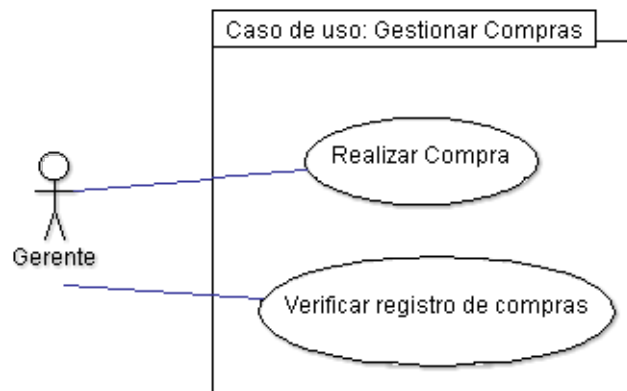
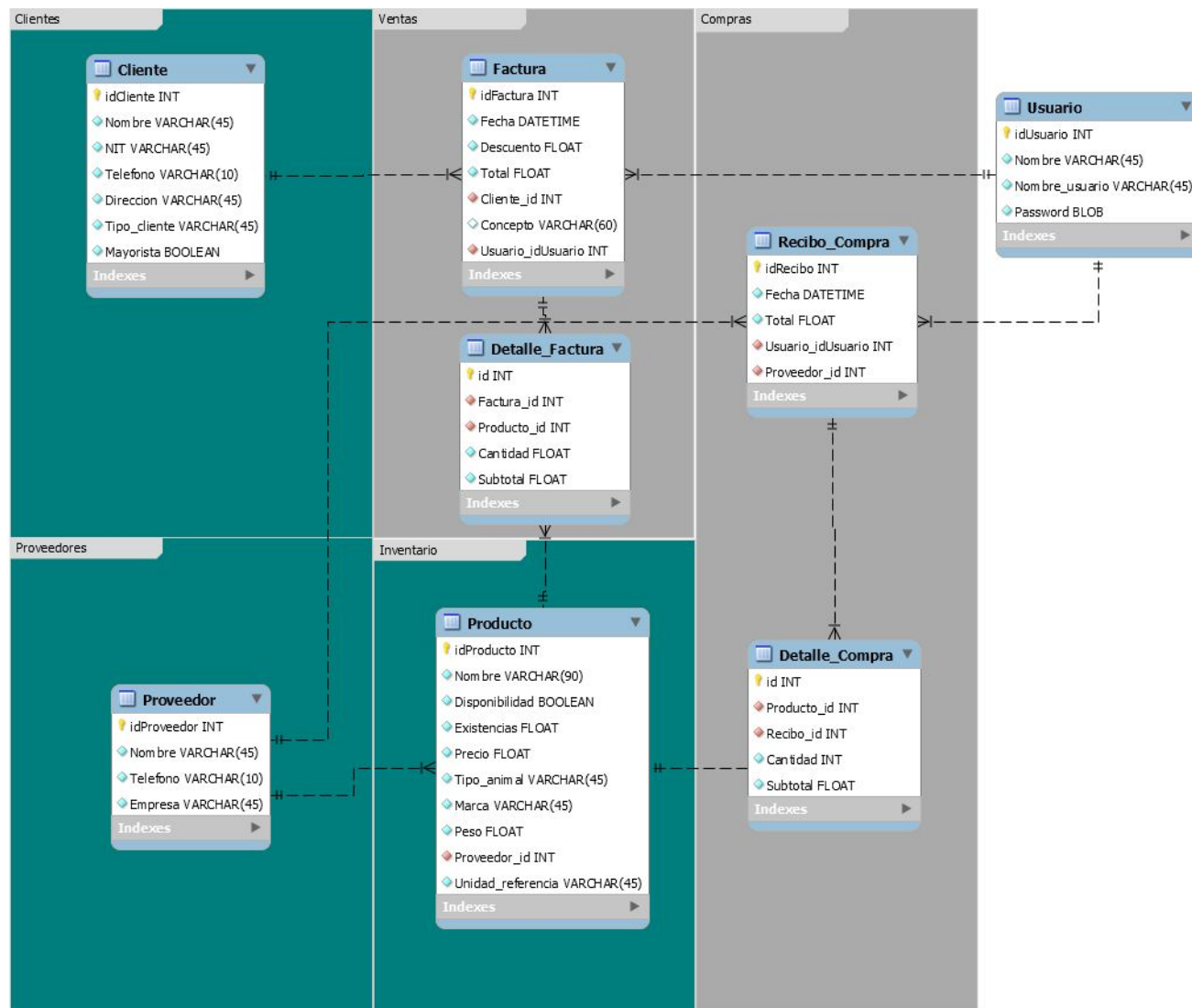
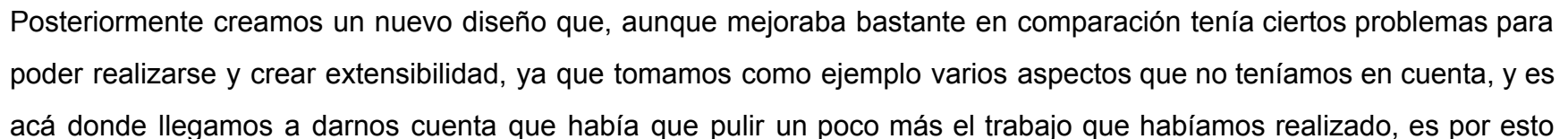


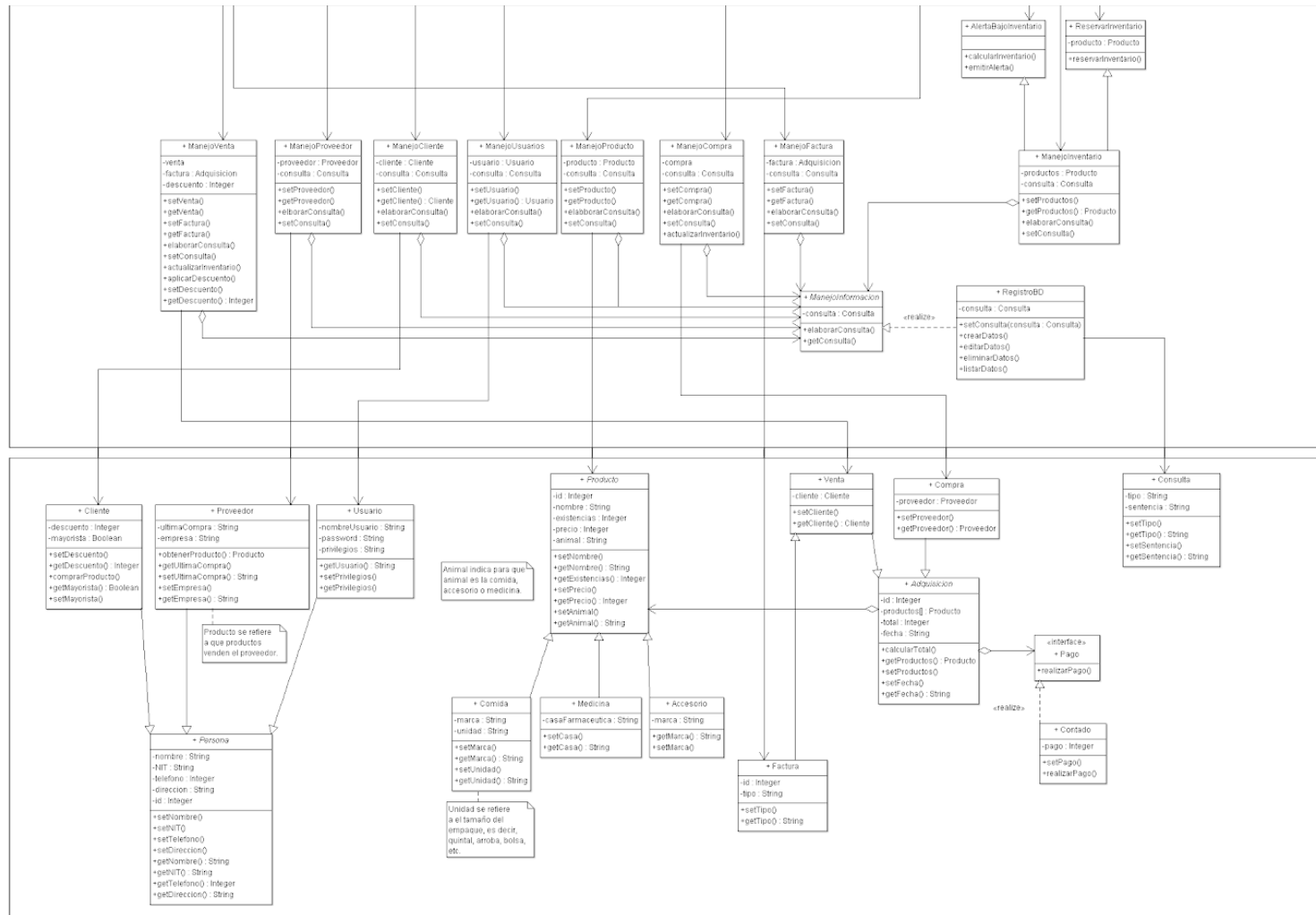
Diagrama Entidad Relación.



En entregas anteriores creamos un diagrama de clases, es el que se muestra a continuación, pero se puede ver que hay muchas clases ejecutan las mismas operaciones y que hacen que el diseño y la cantidad de clases crezca mucho, y que al momento de crear una extensibilidad se vuelva extremadamente complicado.



que rediseñamos, un poco, el diagrama, ya que habían aspectos que habíamos tomado en cuenta, pero otros que no, es por esto que decidimos reestructurar nuestro diagrama, quedando de la siguiente manera:



Por cuestiones de espacio se omitió la capa de la vista, que solamente tiene la definición de clases de la vista y su vinculación con las demás clases.

Pensando en extensibilidad

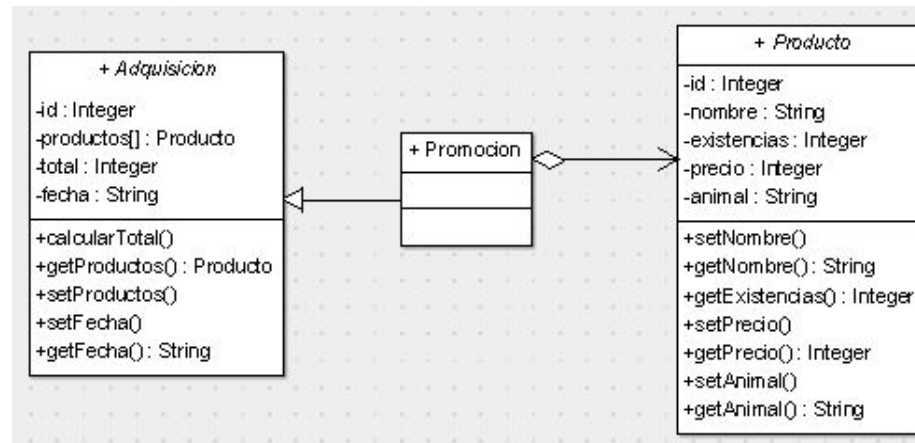
Es sumamente importante el saber que hay varios aspectos que el sistema no contempla, esto debido a que el modelo de negocio tampoco, pero dejamos en claro que si lo tomamos en cuenta, y acá dejamos la evidencia de que si se llegara a pedir su implementación sería muy fácil, y esto haría que no se toquen las clases ya existentes.

*En los siguientes diagramas únicamente se colocarán las clases e interfaces, pero sin métodos o características, ya que estos dependerán de los requisitos que nos indique el dueño de la empresa.

Ofertas

El negocio actualmente trabaja con precios fijos, pero el negocio empieza a crecer y muchos clientes piden ofertas, el dueño les habla a los proveedores y estos acceden, y el negocio está listo para empezar a trabajar, pero, ¿y el sistema?

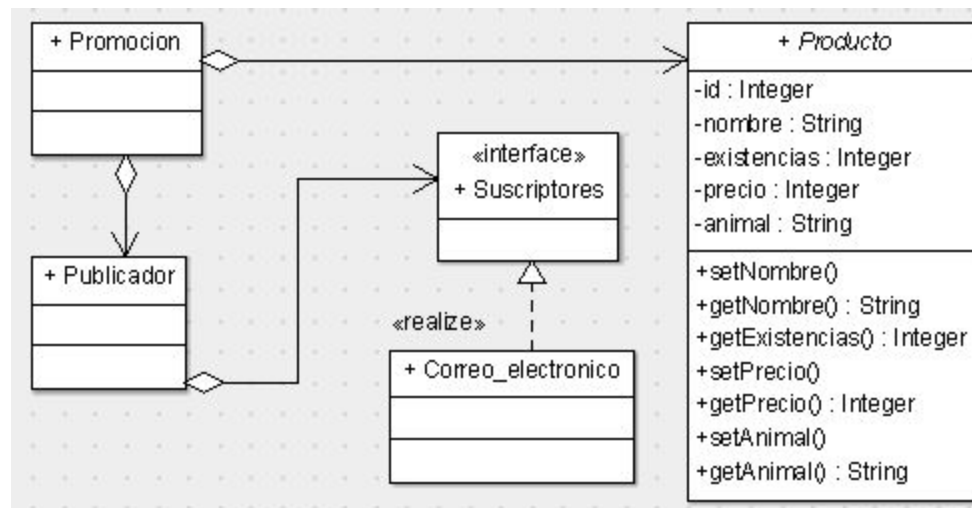
Esto es muy fácil, simplemente se tendría que hacer lo siguiente:



Una simple agregación de la clase promoción sería lo que tendríamos que hacer, y esta a su vez, conectar adquisición con este, que permita la ejecución de ofertas, en este caso la clase “Adquisición” trata a la clase “Promoción” Como un nuevo producto

Notificaciones a clientes

Este va muy relacionado con el punto anterior, ya que queremos comunicar a quienes son nuestros clientes qué ofertas tenemos y queremos que ellos estén informados, esto, como todo, se puede realizar fácil y rápidamente, tal y como sigue.

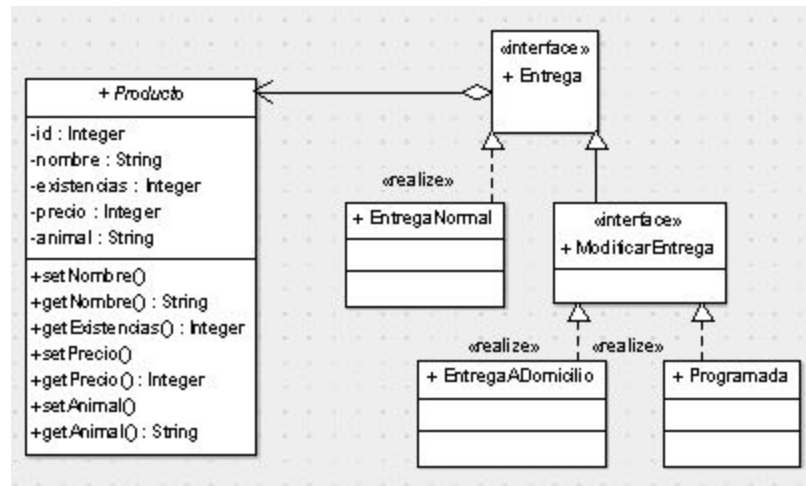


Dirigido por el patrón observador, que hace que el publicador pueda enviar notificaciones a todas las implementaciones de las clases, y esto permite que se pueda notificar por diferentes medios, esto creando una nueva estrategia.

Entregas a domicilio

Actualmente el negocio realiza entregas al momento, en el local, y no permite realizar entregas a domicilio o programar entregas en un tiempo específico.

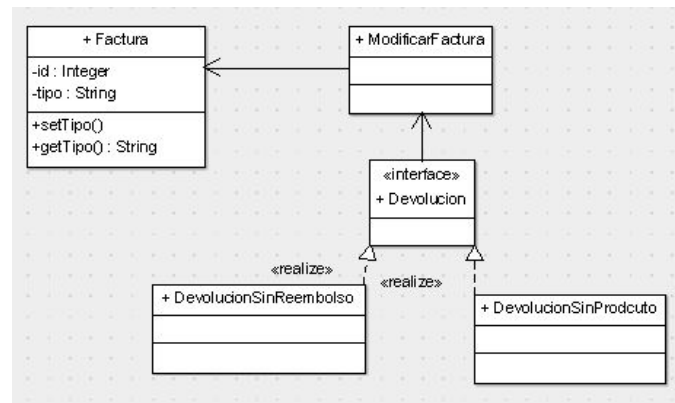
Y quedaría más o menos de la siguiente manera:



Rápidamente podemos ver que se usa un patrón decorador que permite realizar las entregas a domicilio y realizarlas programadas.

Devoluciones

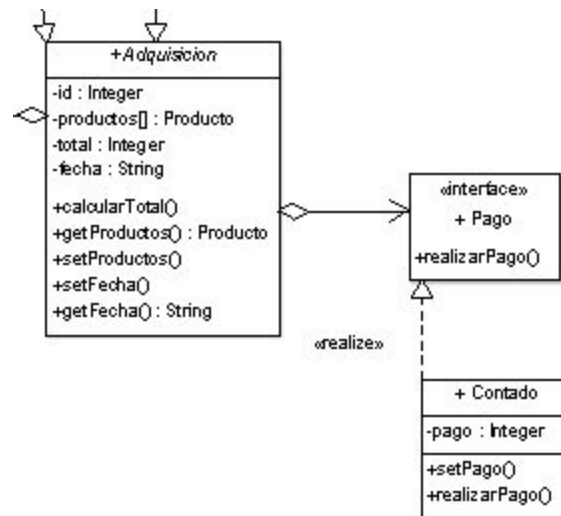
Generalmente no se realizan devoluciones, pero si la empresa crece va a haber más movimiento de productos, en algún momento pueden salir algunos con defectos o en mal estado, esto se puede componer de la siguiente manera:



Esto permite realizar más tipos de devoluciones, y como en todo, va a depender del dueño que nos indique.

Pagos

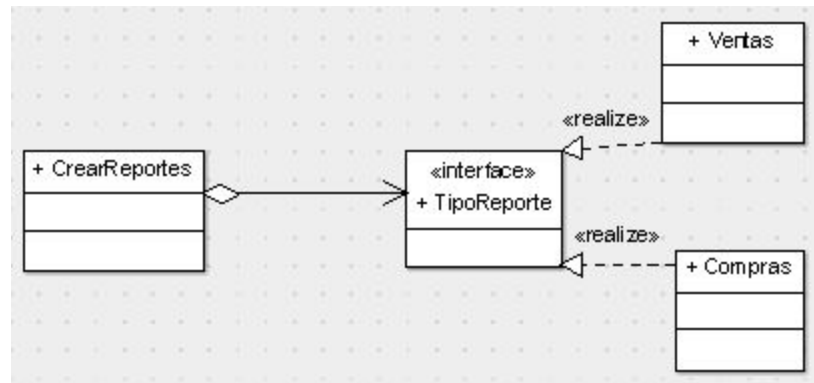
Al momento todos los pagos se realizan al contado, y sin ningún problema. Pero pensemos en el caso de que se quieran agregar pagos con tarjeta, con cheque, por transacción, esto es muy abrumador, pero por suerte, lo tomamos en cuenta y se puede ver de la siguiente manera:



Se hace uso de una modificación de strategy, en donde se puede agregar varios tipos de pagos, como se mencionaba anteriormente.

Reportes

El negocio no utiliza reportes ya que sus cuentas son diarias, pero en algún punto puede llegar a necesitarlo, es por eso que también tomamos en cuenta eso, lo que se puede trabajar de la siguiente manera:



En la imagen se puede ver que se crean dos tipos de reportes, pero estos no se limitan a estos únicos dos tipos, sino que se podría extender a cualquier tipo de reporte que se requiera.

¿Por qué no aplicarlo?

Acá se podría hacer una pregunta muy válida, ¿por qué no aplicar lo que se sabe que puede llegar a pasar?

Y la respuesta es muy fácil, no nos lo pidieron, y ningún módulo depende de estas nuevas funcionalidades, sino que son estas las que dependen de las clases ya existentes.

Es posible que en algún caso el cliente nos lo pida, pero al implementarlo puede ser que no lleguemos a cumplir con los requerimientos iniciales, y lleguemos a entregar algo que no se pida. Es por esto que no se necesita o se ve necesario el aplicar todo lo que veamos que pueda llegar a ser requerido.

Link del repositorio

<https://github.com/jpsancheze6/sistemaGUAU>